Contributions to unsupervised visual anomaly detection

Joao Paulo Casagrande Bertoldo

PSL
UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à Mines Paris – PSL

Contributions to Unsupervised Visual Anomaly Detection

Contributions à la Détection Non Supervisée d'Anomalies Visuelles

Soutenue par

João Paulo Casagrande Bertoldo

Le 7 février 2025

École doctorale n°621

Ingénierie des Systèmes, Matériaux, Mécanique, Énergétique

Spécialité

Morphologie Mathématique

Composition du jury :

Nicolas LOMÉNIE Professeur, Université Paris Cité	*Président du jury*
Eva DOKLADALOVA Professeure, ESIEE Paris	*Rapporteur*
Toby BRECKON Professeur, Durham University	*Rapporteur*
Joana FRONTERA-PONS Ingénieure de Recherche, ONERA	*Examinateur*
Samet AKÇAY Ingénieur de Recherche, Intel	*Examinateur*
Etienne DECENCIÈRE Directeur de Recherche, Mines Paris – PSL	*Directeur de thèse*

MINES PARIS | PSL

Acknowledgements

In 2017, a lifetime opportunity was given to me by Mines Paris. I was invited to join an engineering double degree program at this one-of-a-kind school, who provided me with the resources and support that I needed to succeed in my studies, and for that, I am truly grateful.

In 2020, the school, through the hands of David Ryckelynck, again provided me with another pivotal opportunity. Thanks to David, I discovered the world of research by joining the Centre de Matériaux during an internship with Henry Proudhon. This experience sparked my passion for research and inspired me to pursue this PhD.

In 2021, Mines Paris once more gave me a life-changing opportunity. I was offered a PhD position at the Centre de Morphologie Mathématique (CMM), whose financial support and resources allowed me to complete my PhD. I am deeply grateful to them.

I would like to express my gratitude to my supervisors, Etienne Decencière and Santiago Velasco-Forero, for the opportunity that they gave me to work on this project and for their support. I am also grateful to Anne-Marie de Castro, the secretary of CMM, for her invaluable services and for always going above and beyond to ensure the smooth running of CMM with a smile. Her dedication to supporting researchers, including myself, is truly remarkable and appreciated by all.

I would like to extend my heartfelt thanks to the engineering students who I suppervised during their internships at CMM: Sophian Akkari in 2022; David Arrustico, Tetiana Gula, and Zeyu Jiang in 2023; and Louis Hayot and Sahar Assaoui in 2024. They were a pleasure to work with and greatly motivated my journey. I am grateful for their hard work – which concretly contributed to this thesis – and for the opportunity I had to share my experience with them.

In 2023, the Anomalib team from Intel welcomed my project proposal for the Google Summer of Code program. It was a great experience to work with Ashwin Vaidya, Dick Ameln, and Samet Akçay. I am incredibly grateful for their mentorship and for the opportunity to contribute to such an innovative open-source project.

Finally, I am grateful for the jury members, Eva Dokladalova, Toby Breckon, Joana Frontera-Pons, and Samet Akçay, for their time and commitment to evaluating this thesis.

Abstract

This thesis addresses visual anomaly detection, focusing on unsupervised defect identification in industrial inspection applications. Motivated by recent advances in the field, this work presents contributions to (1) the evaluation standards of anomaly localization, (2) usability of models via post-processing techniques, and (3) model-specific improvements. First, we introduce AUPIMO, a novel evaluation metric that addresses limitations of existing benchmarks. It imposes a hard minimization of false positives on normal samples, encouraging a more challenging and trustworthy evaluation. Experiments across 27 datasets and eight state-of-the-art models demonstrate that AUPIMO provides a more reliable and detailed performance assessment than previous benchmarks. Second, we propose an unsupervised, image-specific threshold selection method for anomaly localization. This method avoids biases introduced by statistics-based thresholds, offering an alternative to color-coded heatmaps. Third, we analyze Gaussian distribution-based models through a novel dimensionality reduction method. This analysis leads to findings that challenge prevailing notions in the field, such as the correlation between variance and model performance. This novel subspace-based dimensionality reduction method, combined with synthetic anomalies, is shown to consistently improve performance. Additionally, we introduce a visualization tool enabling anomaly localization for image-wise Gaussian models. The thesis also presents an incremental improvement to a pixel-wise one-class classification model, enabling more effective use of pixel-wise annotations and faster training. The proposed contributions aim to bridge the gap between research and real-world applications, offering model-agnostic solutions to improve benchmarking and model usability, and model-specific improvements to Gaussian-based models.

Keywords: Visual Anomaly Detection, Unsupervised Learning, Evaluation Metric, Anomaly Localization, Gaussian Models, Dimensionality Reduction, Benchmark, Industrial Inspection, AUPIMO

Résumé

Cette thèse traite de la détection d'anomalies visuelles, en particulier appliquée à l'identification non supervisée de défauts en applications d'inspection industrielle. Motivé par les récentes avancées dans le domaine, ce travail présente des contributions (1) à d'évaluation de la localisation des anomalies, (2) à la facilité d'utilisation des modèles via des techniques de post-traitement, et (3) à des améliorations spécifiques à des modèles basés sur les distributions gaussiennes. D'abord, nous présentons AUPIMO, une nouvelle mesure d'évaluation qui s'attaque aux limites des *benchmarks* actuels. Elle impose une minimisation stricte des faux positifs sur les échantillons normaux, encourageant ainsi une évaluation plus difficile. Des expériences menées sur 27 ensembles de données et huit modèles de l'état de l'art démontrent qu'AUPIMO fournit une évaluation plus fiable et plus détaillée des performances. Deuxièmement, nous proposons une méthode de sélection non supervisée de seuil et spécifique à l'image pour la localisation des anomalies. Cette méthode évite des biais introduits par les seuils basés sur des statistiques et offre une alternative aux *heatmaps* codées en couleur. Troisièmement, nous analysons les modèles basés sur distributions gaussiennes à l'aide d'une nouvelle méthode de réduction de la dimensionnalité. Cette analyse aboutit à des résultats qui remettent en question des notions dominantes dans le domaine, telles que la corrélation entre la variance et la performance en détection d'anomalies. Cette nouvelle méthode de réduction de la dimensionnalité basée sur les sous-espaces, combinée à des anomalies synthétiques, permet d'améliorer les performances de manière consistante. Nous présentons également un outil de visualisation permettant de localiser les anomalies pour les modèles gaussiens conçu pour des images entières. La thèse présente également une amélioration incrémentale d'un modèle de classification à une classe, permettant une utilisation plus efficace des annotations au niveau des pixels et un entrainement plus rapide. Les contributions proposées visent à combler l'écart entre la recherche et les applications du monde réel, en offrant des solutions agnostiques aux modèles pour améliorer l'analyse comparative et la facilité d'utilisation des modèles, ainsi que des améliorations spécifiques aux modèles basés sur les gaussiennes.

Mots clés : Détection d'Anomalies Visuelles, Apprentissage Non Supervisé, Métrique d'Évaluation, Localisation d'Anomalies, Modèles Gaussiens, Réduction de Dimensionnalité, *Benchmark*, Inspection Industrielle, AUPIMO

Contents

Chapter 1

Visual Anomaly Detection

Abstract

This chapter is an introduction to the thesis, presenting the visual anomaly detection task and the field's literature. The chapter provides an overview of public datasets used to benchmark models on different application settings. A special focus is given to MVTec-AD and VisA, which emulate industrial inspection scenarios. It reviews common principles used in deep learning-based methods, including likelihood estimation, one-class classification, reconstruction, and distillation. Finally, the chapter outlines the organization and contributions of the thesis.

Résumé

Ce chapitre est une introduction à la thèse, présentant la tâche de détection d'anomalies visuelles et la littérature dans ce domaine. Le chapitre donne un aperçu des ensembles de données publiques utilisés pour comparer les modèles de détection d'anomalies visuelles dans différents contextes d'application. Une attention particulière est accordée à MVTec-AD et VisA, qui émulent des scénarios d'inspection industrielle. Différents principes de détection utilisés dans les méthodes basées sur l'apprentissage profond sont passés en revue, y compris l'estimation de la vraisemblance, la classification en une classe, la reconstruction et la distillation. Enfin, le chapitre présente l'organisation et les contributions de la thèse.

Contents

1 Introduction

Anomaly detection in machine learning aims to identify patterns that deviate significantly from the norm – the *anomalies*. The normal samples, patterns, or events are assumed to be abundant enough in the training set so its variations can be learned. The key difference with classification tasks is that one cannot define a priori what an anomaly is. Thus, from a learning perspective, it is not feasible to characterize anomalies as a class – although, in practice, they are often referred to as such.

From an application perspective, the normal patterns are often not of special interest at inference time. The anomalies are the users' focus, and they are commonly associated with events having negative consequences or deserving special attention/inspection. A related research field is novelty detection, where this negative connotation is not (or less) present. In both cases, the inference consists of identifying unseen patterns, which can manifest in countless ways. In this thesis, we do not differ between the two, but the datasets in our experiments are from an industrial inspection context (details in Section 2), so we refer to the task as *anomaly* detection. Other examples of applications encompass various domains, including:

- Financial sector: identifying fraudulent transactions of individuals or groups of individuals;
- Computer system monitoring: detecting unusual latency, error rates, logs, etc;
- Network security: highlighting unusual traffic patterns indicative of a cyberattack;
- Medical imaging: identifying tumors, abnormalities, or unrecognized pathologies;
- Surveillance: identifying suspicious activities (*e.g.* trespassing, unattended baggage);
- Autonomous driving: detecting unexpected objects or behaviors on the road.

As a machine learning task, anomaly detection carries an interest in its own. It is a challenging problem due to the lack of contrast between normal (*i.e.* known) and anomalous (*i.e.* unknown) patterns. The lack of definition of what an anomaly is makes it hard to define a loss

function that can be optimized. This challenge has led to the development of various methods, covering a wide range of approaches (details in Section 3).

Other related problems include open set recognition, out-of-distribution detection, and outlier detection. These problems have their own characteristics/objectives – which have been discussed in the litterature [He, 2003; Yang, 2024; Salehi, 2022][1] – but their common thread is the unsupervised nature of identifying uncategorized patterns. In other words, the lack of a priori definition of what one is looking for. The key difference is that, in anomaly detection, the normal class is well defined by the training set. The latter is assumed to be representative of what users consider to be normal variations, thus not carrying any "interesting" or "surprising" information.

When the training is assumed to have *only* normal samples/patterns (*i.e.* a "clean" dataset), the learning task is referred to as a *semi-supervised* because there normal samples are *certain* to be normal [Xu, 2023; Ruff, 2021a]. In practice, this assumption is not always true, especially in real-world scenarios. The training set might be contaminated with anomalous variations due to mislabeling, noise, or other factors. In such cases, the anomaly detection is referred to as *unsupervised*, and the learning model must be robust to the presence of contaminated samples.

This aspect is not directly addressed here because the datasets used in this thesis are "clean", but other authors have tackled it [Xu, 2023; Etherington, 2021; Leys, 2018; Wang, 2023]. In this context, our focus is on semi-supervised anomaly detection, but part of the litterature still refers to it as unsupervised due to the lack of anomaly samples at training time. A special focus is given to this unsupervised nature at inference time in Chapters 2 and 3.

This thesis focuses on *visual* anomaly detection in the context of industrial inspection with 2D images[2]. The term "[anomaly] *detection*" in this context typically refers to the *image-wise* detection task, where the objective is to determine whether an image contains an anomalous structure. The term "image-wise" is used to emphasize that the model outputs a single score for the entire image. The ground truth annotation is a single binary label (0 for normal, 1 for anomalous).

A closely related task is "anomaly *localization*" (or "anomaly *segmentation*"), which aims to pinpoint the specific pixels within an image that belong to the anomaly. Analogously, the term "pixel-wise" is used to emphasize that the model outputs a score for each pixel in the image, and the ground truth annotation is a binary mask with ones indicating the anomaly's location. These pixel scores are arranged in a 2D heatmap, often superposed to the input image, where higher values indicate a higher belief/confidence that a pixel is anomalous (see Fig. 2.1 for an example). Most models are designed to produce pixel-wise scores, which are then aggregated to produce an image-wise score (*e.g.* by taking the maximum or sum of the pixel-wise anomaly scores).

Anomalies are often assumed to be rare and hard to obtain. This association led to the common practice of defining anomalies as instances with a low probability. However, in the

[1] These topics are *not* covered in this thesis, but the provided references convey a good overview of the state of the art and the differences between these problems.

[2] The field of visual anomaly detection also encompasses other modalities such as video, 3D images (*e.g.* X-ray tomographies), and point clouds (*e.g.* Lidar data).

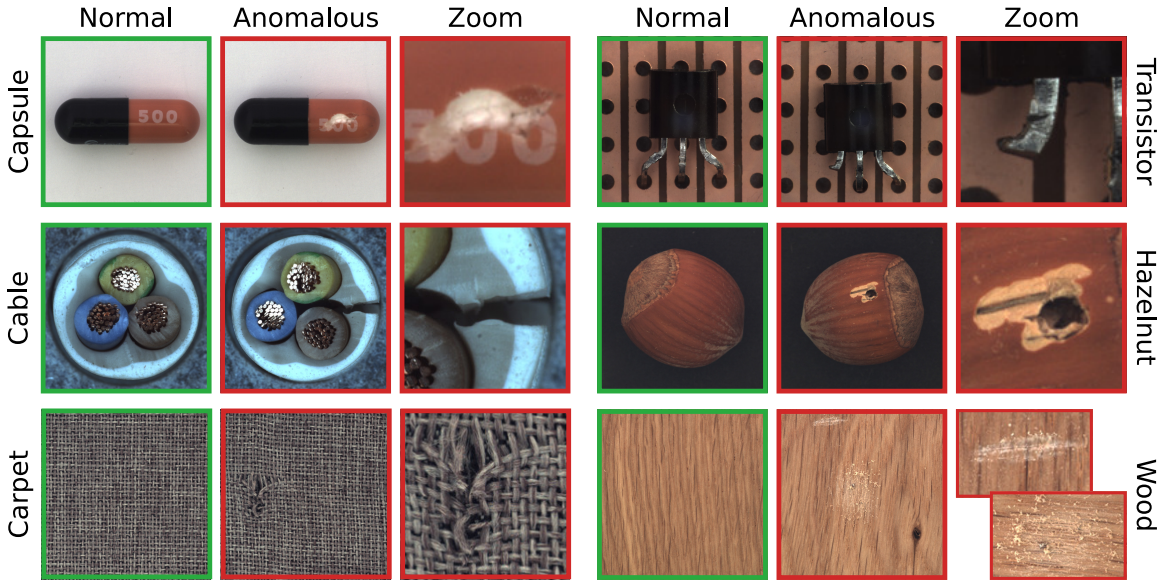


Figure 1.1: Samples from the MVTec AD dataset. More samples can be seen in Fig. 2.5.

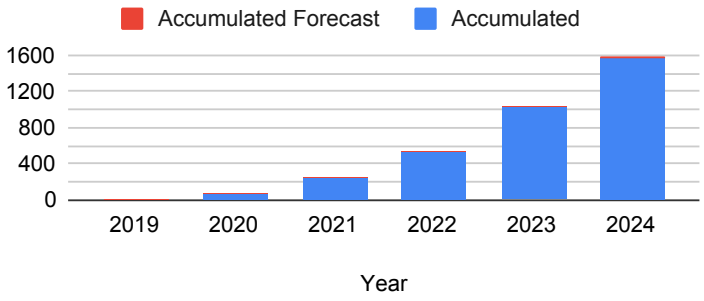


Figure 1.2: Citations of the MVTec AD dataset. Number of citations per year accumulated from left to right. Source: Google Scholar.

visual domain this association is not necessarily correct. As an example, consider a hypothetical road video surveillance application. In the United States, only one in a thousand cars is yellow [GermainCars, 2024], but should a yellow car be considered an anomaly in this context? Probably not because the color of a car doe not show any worth-inspecting behavior/information. This distinction highlights a pragmatic consideration: the training dataset must encompass the full range of normality. For the rest of this thesis, we assume that the training set is representative of all normal variations, although consideration should be given to the context of practical applications.

In the rest of this chapter, we present a brief overview of the field of visual anomaly detection. Section 2 presents public datasets used in visual anomaly detection. A special attention is given to MVTec Anomaly Detection (MVTec-AD) and Visual Anomaly (VisA), which emulate industrial inspection scenarios, because they were extensively used in our experiments. Section 3 reviews relevant methods and topics in the litterature. The goal is to give an overview of a wide range of approaches as a motivation for the contributions in this thesis. In particular, Chapters 2 and 3 present model-agnostic contributions, which are applicable to most of those methods.

2 Datasets

Historically, the field of visual anomaly detection has relied on standard image datasets such as MNIST [Lecun, 1998], Fashion-MNIST [Xiao, 2017], CIFAR [Krizhevsky, 2009], and ImageNet [Deng, 2009]. These datasets have been used in the one-vs-rest setting, where one class is treated as normal, and the rest are treated as anomalies. However, this setup does not capture the complexities of real-world applications. They often have classes that are easy to distinguish and do not challenge current models to learn the subtleties of the normal class.

To address this limitation, specialized benchmark datasets have been developed for different applications and domains. We present a list covering most of the datasets used in the papers we cite throughout this thesis (per application):

- Industrial inspection:
  - MVTec Anomaly Detection (MVTec-AD) [Bergmann, 2019; Bergmann, 2021a];
  - MVTec-C [Wang, 2023], an extension of MVTec-AD with common image corruptions (motion and defocus blur, light and contrast issues, compression, etc);
  - MVTec logical constraints anomaly detection dataset (MVTec LOCO AD) [Bergmann, 2022];
  - MVTec 3D AD [Bergmann, 2021b], with 3D point clouds;
  - Visual Anomaly (VisA) [Zou, 2022];
  - beanTech Anomaly Detection dataset (BTAD) [Mishra, 2021];
  - Printed Circuit Board (PCB) [Huang, 2019];
  - Magnetic Tile Defect [Huang, 2018];
  - metal parts fabrication [Jezek, 2022];
  - Photovoltaic Cell Electroluminescence Anomaly Detection (PVEL-AD) [Su, 2023];
  - Supermarket Goods Anomaly Detection (GoodsAD) [Zhang, 2023a];
  - Vision-based InduStrial InspectiON (VISION) [Bai, 2023];
  - Kolektor Surface-Defect Dataset (KolektorSDD) [Tabernik, 2020] and KolektorSDD2 [Božič, 2021];
  - Photometric Stereo Anomaly Detection (PS-AD) [Jung, 2022], where each object sample has several images taken under different lighting conditions;
- Autonomous driving: Road Anomaly [Lis, 2019]; Fishyscapes [Blum, 2021]; BDD-Anomaly [Hendrycks, 2022], which adapts the BDD100K dataset [Yu, 2020] for anomaly detection by combining it with simulation-generated anomalies;
- Medical imaging: histopathological images [Zingman, 2022]; brain Magnetic Resonance Imaging (MRI) and abdominal Computed Tomography (CT) images [Zimmerer, 2022]; Benchmarks for Medical Anomaly Detection (BMAD) [Bao, 2023], which gathers six datasets;
- Video surveillance: Shanghaitech Campus [Luo, 2017; Liu, 2018]; UCSD Pedestrian datasets [Mahadevan, 2010]; Street Scene [Ramachandra, 2020]; A Day on Campus (ADOC) [Pranav, 2020]; CUHK Avenue [Lu, 2013].

In particular, MVTec-AD (see Fig. 1.1) has become a standard benchmark for evaluating visual anomaly detection models thanks to its realistic industrial inspection scenarios and diverse set of anomalies. It comprises 15 categories, each representing a different object or texture commonly encountered in manufacturing processes. Each category in MVTec-AD is an independent dataset, with a pre-defined separation between training and testing sets. The training set consists of only defect-free (normal) images, with usually around 200-300 images per dataset. The testing set includes both normal (generally 20-30 images) and anomalous images (around 100-150 images) for each dataset. The anomalies manifest in the form of different types of defects such as scratches, dents, contaminations, and various structural changes. The datasets have high-resolution (betwee 700×700 and 1024×1024) color and gray images, and provide pixel-precise ground truth annotations for all anomalies. Its focus on structural anomalies and the availability of both image-wise and pixel-wise annotations make it a valuable resource for assessing the performance of anomaly detection and localization methods (details in Chapter 2).

VisA is another important benchmark for visual anomaly detection, also focusing on industrial inspection scenarios. It similarly provides 12 independent categories with high-resolution images and pixel-wise anomaly annotations, making it suitable to complement MVTec-AD by offering a more diverse set of challenges. In this thesis, we utilize MVTec-AD and VisA for our experiments due to their relevance in the field.

3 Inference Principles

In recent years, deep learning-based methods have dominated the literature due to their ability to learn complex representations from high-dimensional data like images. The capacity of neural networks to automatically extract relevant features – eliminating the need for manual feature engineering – has made them particularly well-suited for visual anomaly detection tasks. The approaches seen in the litterature are diverse and differ substentially from tasks like classification or regression. In this section, we review common principles used in deep learning-based visual anomaly detection methods.

The categorization proposed here is based on the inference principle, *i.e.* how a model determines the anomaly score of a sample. We categorize the methods into four main groups: likelihood estimation, one-class classification, reconstruction, and distillation. Notice that our categorization that does not cover aspects of the learning process. Our goal is not to propose a definitive taxonomy, but rather to provide a high-level overview of the approaches in the field.

Several reviews encompassing a broader range of methods have been published in the literature [Raghavendra, 2019; Pang, 2021a; Pang, 2021b; Ruff, 2021a; Tao, 2022; Cordier, 2022; Wang, 2020b; Han, 2022; Hojjati, 2024; Cui, 2023]. They have used other taxonomies with different categorization criteria. The field's diversity makes it challenging to create a universal taxonomy because many models mix multiple techniques. For example, self-supervised models often minimize a loss function during training but employ, for instance, a nearest neighbors-based approached using the learned feature space for inference.

Our review is not exhaustive, but the chosen methods are representative of the state of the art

in visual anomaly detection and popular topics in the recent litterature. A considerable number of relevant papers do not focus on the inference principle, but rather on a specific component of the learning process (*e.g.* the self-supervised loss function in the example above). Therefore, the last category (Section 3.5) encompasses component-focused papers that generally use one of the four inference principles mentioned above.

This large diversity in the visual anomaly detection field motivates the model-agnostic contributions in the following chapters. Chapter 2 focuses on establishing a solid evaluation framework for future research, and Chapter 3 addresses a common issue in anomaly localization methods proposing a model-agnostic solution.

3.1 Likehood Estimation

Anomalies are commonly associated with unlike events. This intuition is the basis of likelihood estimation methods, where the goal is to detect samples unlikely to be generated by from the distribution of normal samples. The observed normal samples are used to estimate this distribution, which is used to assess the likelihood of a new sample at inference time. Data points with low likelihood are considered anomalous.

We split this category into two groups: feature extraction and network optimization. "Feature Extraction" refers to methods that use a (generally pre-trained) neural network to extract dense vectors from the input images. This alternative representation of the images is then combined with another method to estimate the likelihood. "Network Optimization" refers to methods that train a neural network – *i.e.* update its parameters – to directly estimate the likelihood.

Feature Extraction These methods typically use a neural network pre-trained on a large classification dataset like ImageNet [Deng, 2009] or a self-supervised task. The internal activations of the network – the features – are extracted and used to encode the information in the images in a vector space. These feature vectors are then used as a representation of the images to estimate the likelihood.

A simple approach is to assume that the normal samples follow a Gaussian distribution [Rippel, 2021c; Rippel, 2021a; Rippel, 2021b; Etherington, 2021; Leys, 2018; Lin, 2022; Tschuchnig, 2022; Kamoona, 2024]. The training step consists of estimating the mean and covariance matrix of the Gaussian, which allows one to analytically compute the likelihood of a sample. At inference time, the likelihood is often replaced by the Mahalanobis distance. The Mahalanobis distance is proportional to the negative log-likelihood of a sample, so samples with a high Mahalanobis distance (*i.e.* "far" from the mean vector) are considered anomalous. [Rippel, 2021c; Rippel, 2021a; Rippel, 2021b; Lin, 2022; Kim, 2021] empirically observed that reducing the dimensionality of the feature space by eliminating high-variance principal components can improve the anomaly detection performance. The first part of Chapter 4 (Section 2), largely based on [Gula, 2023; Jiang, 2023], analyzes this phenomenon in depth and reaches a different conclusion.

PaDiM (Patch Distribution Modeling Framework) [Defard, 2021] estimates one Gaussian distribution per pixel position in the feature maps (*i.e.* a 3D activation tensor). This method

became particularly popular thanks to its light weight and localization-enabled solution, which works well for industrial inspection scenarios because images are often aligned. Several follow up works have explored extensions of this method [Kim, 2021; Wan, 2022a; Li, 2021b; Zheng, 2022; Rippel, 2022; Dini, 2022; Jang, 2023]. For instance, [Kim, 2021] generalized their dimensionality reduction method, [Zheng, 2022] included an image alignment step, and [Jang, 2023] combined information from neighboring distributions. The second part of Chapter 4 (Section 3) relates to PaDiM focusing on the visualization of the feature space – the first part exploring a similar model, based on [Bertoldo, 2023b].

The Gaussian assumption limits the capacity to learn, for example, multimodal distributions. A non-parametric alternative is to store the feature vectors of normal samples in a memory bank and use a nearest neighbors search to estimate the likelihood of a new sample [Cohen, 2020; Roth, 2022; Yi, 2020; Zhang, 2022b; Bergman, 2020; Zhang, 2022c; Kim, 2022a; Bae, 2022; Jiang, 2024; Galesso, 2023]. The likelihood is proxied by the distance to the nearest neighbors (higher distance implies lower likelihood).

SPADE (Semantic Pyramid Anomaly Detection) [Cohen, 2020] and PatchCore [Roth, 2022] became particularly popular. SPADE uses a two-level nearest neighbors search, first at the image then at the pixel level. PatchCore directly operates at the pixel level and it uses coreset [Agarwal, 2006] to reduce the memory bank by selecting a subset of the patch-wise feature vectors from the normal samples. While conceptually simple, PatchCore is effective and beat the state of the art on MVTec-AD when it was published (*cf*. Fig. 2.1). Extensions of this method have been proposed: [Jiang, 2024] introduced an image registration step to align the patches, and [Bae, 2022] modified the feature vectors to include information from pixel position and local context (neighboring patches).

Network Optimization These methods incorporate the likelihood estimation into the neural network itself. Instead of relying on post-network density estimation techniques, the network outputs the likelihood directly, and its parameters are adjusted to maximize the likelihood of the observed normal samples. At inference time, the likelihood of a new sample is directly computed by the network. We present two major approaches: energy-based models and normalizing flows.

Energy-based models (EBM) [Dehaene, 2020; Lafon, 2023; Genc, 2021] consist of learning an energy function E_θ, a neural network with parameters θ. Given an input $\mathbf{x} \in \mathbb{R}^D$, it assigns an energy value $E_\theta(\mathbf{x})$, which parametrizes a density function $p_\theta(\mathbf{x})$ such as [Lecun, 2006]

$$p_\theta(\mathbf{x}) = \frac{\exp(-E_\theta(\mathbf{x}))}{\int_{\mathbb{R}^D} \exp(-E_\theta(\mathbf{x}))\mathrm{d}\mathbf{x}} \quad .$$

Here, the output of the network for a sample corresponds to the negative log-likelihood, so higher energy indicates lower likelihood. The parameters θ are adjusted to minimize the energy of normal samples, thus maximizing the likelihood of the normal class. At inference time, the energy value of a new sample serves as its anomaly score. The energy function can be seen as a measure of how compatible a sample is with the learned model of normality, encoded in the network.

Normalizing flows [Papamakarios, 2021] are a more popular alternative in the visual anomaly detection literature [Gudovskiy, 2022; Kim, 2022b; Yu, 2021; Rudolph, 2022; Lei, 2023; Rudolph, 2021; Tailanian, 2024]. These models learn a transformation that maps a data point – which can be the image or its feature maps – coming from a complex distribution to a simple one. The base assumption is that a normal instance $\mathbf{x}$ is a transformation $\mathbf{x} = T_\theta(\mathbf{u})$ of a real vector $\mathbf{u}$ of the same dimension. The latter is distributed according to a (known and tractable) base distribution p_u (typically, a standard Gaussian). The transformation T_θ is a neural network (parametrized by θ) designed to be invertible, enabling the computation of the likelihood of $\mathbf{x}$ from $T_\theta^{-1}(\mathbf{x})$. During training, the network parameters are adjusted to maximize the likelihood of the observed normal samples or to minimize the Kullback-Leibler divergence between the transformed distribution and a target distribution.

CFLOW-AD [Gudovskiy, 2022] and FastFlow [Yu, 2021] are particularly popular, which were followed by more recent works like PyramidFlow [Lei, 2023] and UFlow [Tailanian, 2024]. PyramidFlow has a multi-scale design where $\mathbf{u}$ has multiple resolutions corresponding to different layers from a convolutional neural network. UFlow goes further by upsampling information from the lower resolution layers to the higher ones in the transformation network – a design inspired by the U-Net architecture [Ronneberger, 2015a].

3.2 One-Class Classification (OCC)

One-class classification (OCC) models detect anomalies by learning a compact representation of the normal data. Early OCC methods [Schölkopf, 2001] like Support Vector Data Description (SVDD) [Tax, 2004] create a boundary using a hypersphere to enclose the normal data. Deep SVDD [Ruff, 2018] built on this idea by training neural networks to minimize the distance of normal instances to the center of a hypersphere in the feature space. At inference time, the distance of a sample to the center of the hypersphere serves as its anomaly score (further from the center implies lower similarity to the normal class). However, this approach can suffer from feature collapse (*i.e.* the network learns to map all samples to the same point). To address this, extensions like the Hypersphere Classifier (HSC) loss incorporate anomalous samples (or outlier exposure, *i.e.* an unrelated dataset) to push them away from the center, improving boundary definition [Ruff, 2020; Liznerski, 2022]. Several other extensions have been proposed [Goyal, 2020; Zhang, 2022d; Lee, 2022; Massoli, 2022], among which an adaptation modeling patch-wise vectors [Liznerski, 2021] is discussed in Chapter 5. An extensive review of OCC models can be found in [Perera, 2021].

3.3 Reconstruction

Reconstruction-based methods generally transform the input data into another representation and then attempt to reconstruct the original input from it. They operate on the principle that a model trained to accurately reconstruct normal samples will struggle to do the same for anomalies. Consequently, deviations from the learned normal patterns, as would be present in an anomaly, lead to reconstruction errors. This error serves as the anomaly score, with higher

errors implying that the model's transformations do not recognize the input's structure.

Many reconstruction strategies have been proposed [Lis, 2019; Zavrtanik, 2021b; Imamura, 2021; Liu, 2021; Madan, 2022; Ndiour, 2022; Zhou, 2022a; Schwartz, 2022; Battikh, 2022; Samele, 2022]. We mention a few examples to illustrate the diversity of ideas in the litterature. In [Lis, 2019], a network synthesizes the input image from a semantic segmentation map of the same image. In [Ndiour, 2022], a Principle Component Analysis (PCA)-based truncation is applied to the features extracted by a pre-trained network. Similarly, in [Samele, 2022], a Sigular Value Decomposition (SVD) is applied to the feature maps to reduce their dimensionality. A common strategy is to directly learn to reconstruct normal images by ignoring synthetically added anomalies [Wang, 2020a; Zavrtanik, 2021b; Astrid, 2021]; DRAEM (discriminatively trained reconstruction anomaly embedding model) [Zavrtanik, 2021a] is a popular example of this approach.

Autoencoders Autoencoders have been largely explored in the anomaly detection literature [Zhou, 2020; Sakurada, 2014; Venkataramanan, 2020; Shi, 2021; Yang, 2021; Zhou, 2022b; Battikh, 2022; Zimmerer, 2019; Salehi, 2021a; Yao, 2022; Jonak, 2022]. These neural networks consist of two main components: an encoder and a decoder. The encoder maps the input data into a lower-dimensional latent representation, compressing the input information. The decoder then reconstructs the input from this latent representation. During training, the autoencoder learns to minimize the reconstruction error only for normal samples. At inference time, a higher reconstruction error for a given sample suggests that it deviates from the learned normal patterns, flagging it as a potential anomaly. The underlying assumption is that the network tends to overfit to the training data, leading to poor generalization to unseen patterns.

Prototypes Prototype-based methods offer an alternative approach to reconstruction by learning a set of representative prototypes that capture the essence of the normal data [Jezequel, 2022; Zhang, 2022a; Park, 2020; Gong, 2019; Gui, 2022]. These prototypes can be seen as "archetypal" examples of normal patterns. Instead of directly reconstructing the input, these methods replace observed samples with learned prototypes, forcing reconstruction errors when the input contains anomalies. This memory bank-based idea also appears as an auxiliar component in other methods.

3.4 Distillation

Distillation-based methods in anomaly detection draw inspiration from knowledge distillation, a technique used to train a smaller (student) network to mimic a larger (teacher) network's behavior. These methods leverage the knowledge acquired by the teacher network, typically trained on a large classification dataset. The student network is trained to reproduce the teacher's predictions only for normal samples. During inference, the discrepancy between the teacher and student predictions serves as the anomaly score. The working assumption is that the student network, having a weaker architecture and having learned only from normal samples, will struggle to reproduce the teacher's activations for anomalous samples.

Several state-of-the-art methods in visual anomaly detection are based on distillation [Bergmann, 2020; Deng, 2022; Cao, 2022; Zhou, 2023; Tien, 2023; Zhang, 2023b; Salehi, 2021b]. Namely, EfficientAD [Batzner, 2024] – based on Uninformed Students [Bergmann, 2020] – combines a student-teacher architecture with an autoencoder. Reverse Distillation (RevDist) [Deng, 2022] and its follow up RevDist++ [Tien, 2023] is another prominent method, which introduces a shared latent space between the teacher and student networks.

3.5 Other

Here we present other contributions from the litterature that did not fit in the previous categories. They generaly focus on a specific component of the learning process, another learning paradigm, or combine multiple techniques. For instance, clustering methods [Chao, 2022; Wan, 2022a] and GANs (Generative Adversarial Networks) [Goodfellow, 2014] have also been explored [Radford, 2016; Schlegl, 2017; Schlegl, 2019; Akcay, 2019; Akçay, 2019]. Other authors have focused self-supervised and contrastive learning to produce a better feature extractor [Li, 2020; Wan, 2022b; Tsai, 2022; Tack, 2020; Reiss, 2021; Wei, 2023; Sohn, 2021]. These papers often present a detailed analysis of the pre-training process, which can be seen as a model-agnostic contribution, and use another method for the inference step (*i.e.* a Gaussian model). Another example of model-agnostic contribution is the use of synthetic anomalies to augment the training data [Mirzaei, 2022; Li, 2021a; Schlüter, 2022]. Synthetic anomalies have been largely discussed because this component is used in many methods, and the diversity and quality of the synthetic anomalies are crucial to the model's performance.

There are yet other worth-mentioning methods that do not fit neatly into the previous categories like WinCLIP [Jeong, 2023], which adapts CLIP [Radford, 2021], a vision-language model, to zero-shot and few-shot anomaly detection. Finally, a recent paper proposed Simplenet [Liu, 2023], achieving state-of-the-art performance with surprisingly simple methodology: they train a binary classifier, where the anomalies are directly synthesized in the feature space.

4 Outline and Contributions

The remainder of this thesis is organized as follows.

Chapter 2 introduces AUPIMO, a novel evaluation metric designed to address limitations of existing visual anomaly detection benchmarks. It provides a more nuanced evaluation in an unbiased and meaningful way. Our benchmark (eight state-of-the-art models, 27 datasets) demonstrates that AUPIMO establishes a more challenging evaluation and showcases the metric's properties.

Chapter 3 challenges the conventional use of color-coded heatmaps for anomaly localization. We propose an unsupervised, image-specific threshold selection method, avoiding the biases introduced by anomaly-dependent statistics. Experiments demonstrate the potential of this approach to improve anomaly segmentation.

Chapter 4 provides a comprehensive analysis of Gaussian distribution-based models. We investigate the relationship between feature space variance and model performance, challenging

a previous empirical observation that low-variance subspaces are inherently more discriminative for anomaly detection. This investigation leads to a novel subspace-based feature reduction method that consistently improves anomaly detection performance. Finally, we introduce a visualization tool to enhance the interpretability and enable anomaly localization for image-wise-designed Gaussian models.

Chapter 5 presents an incremental improvement to a Support Vector Data Description (SVDD)-based model for anomaly segmentation. By modifying the loss function of Fully Convolutional Data Description (FCDD) [Liznerski, 2021], a more effective use of pixel-wise annotations is achieved. This modification leads to consistent performance improvements across various datasets while requiring fewer training epochs.

The overarching goal of this thesis is to advance the field of visual anomaly detection as a whole, both through general contributions and by exploring model-specific improvements. The contributions in Chapters 2 and 3 offer solutions for better model performance assesment and usability, which are applicable to a wide range of models. We believe these contributions will be of particular interest to the research community as they address fundamental aspects of visual anomaly detection. Chapters 4 and 5, on the other hand, focus on specific models. In particular, Gaussian-based approaches are given special attention due to their simplicity, which makes them more accessible to a wider audience. By exploring and extending these models, we contribute to a better understanding of their inner workings and how one can improve their usability and performance.

Chapter 2

Bias-free Metric (AUPIMO)

Abstract

This chapter introduces a novel performance metric for visual anomaly detection called Area Under the Per-IMage Overlap (AUPIMO) which addresses the limitations of existing metrics – generally borrowed from binary classification. AUPIMO is designed to be unbiased towards known anomalies, making it suitable for unsupervised anomaly detection. It establishes a harder challenge for visual anomaly detection by imposing a low tolerance for false positives on normal images. Our experiments demonstrate that AUPIMO is robust to noisy annotations, accelerates computation, and enables detailed performance assessment. We establish a new benchmark for visual anomaly detection, and the results challenge the perception that datasets from MVTec-AD and VisA have been solved by contemporary models. A conference paper version of this chapter was accepted to The 35th British Machine Vision Conference (BMVC 2024) [Bertoldo, 2024].

Résumé

Ce chapitre présente une nouvelle métrique de performance pour la détection d'anomalies visuelles, appelée Area Under the Per-IMage Overlap (AUPIMO), qui s'attaque aux limites des mesures existantes - généralement empruntées à la classification binaire. AUPIMO est conçu pour ne pas être biaisé par les anomalies connues, ce qui la rend adaptée à cette tâche de détection non supervisée. Elle pose un défi plus difficile à relever pour la détection d'anomalies visuelles en imposant une faible tolérance aux faux positifs sur des images normales. Nos expériences démontrent qu'AUPIMO est robuste aux annotations bruitées, qu'elle rapide à calculer et qu'elle permet une évaluation détaillée des performances. Nous établissons une nouvelle référence pour la détection d'anomalies visuelles et les résultats remettent en question la perception selon laquelle les ensembles de données de MVTec-AD et VisA ont été résolus par des modèles contemporains. Une version en format conférence de ce chapitre a été acceptée à la 35th British Machine Vision Conference (BMVC 2024) [Bertoldo, 2024].

Contents

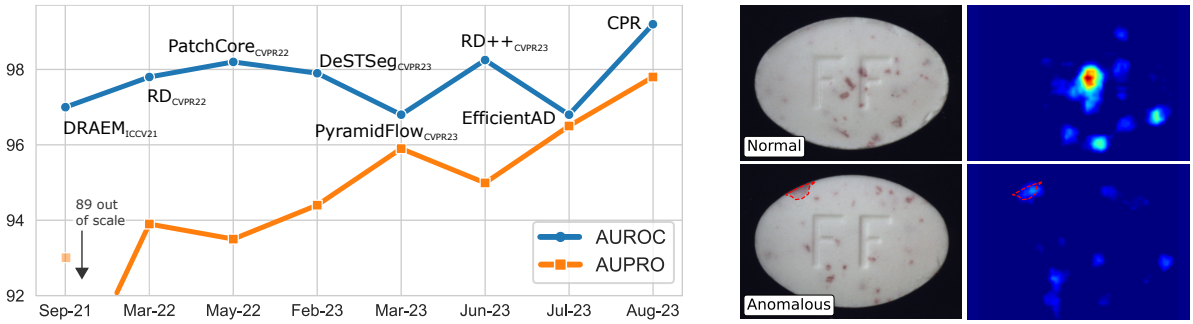


Figure 2.1: Left: performance on MVTec AD over time, approaching a near 100% performance plateau. Right: images from the dataset Pill (left column) and their inferred anomaly maps (right column; higher values mean anomalous; jet colormap) from the best performing model in this dataset (EfficientAD), with 98.7% AUROC and 96.7% AUPRO. The normal image (top) has higher anomaly scores than the anomaly (bottom).

1 Context

Visual anomaly detection (VAD) research has achieved significant progress, partly thanks to the increased availability of suitable datasets [Bergmann, 2019; Zou, 2022; Mishra, 2021; Božič, 2021; Krohling, 2019; Mahadevan, 2010; Pranav, 2020; Ramachandra, 2020]. In particular, the MVTec-AD [Bergmann, 2019; Bergmann, 2021a] and VisA [Zou, 2022] comprise (together) 27 datasets (22 object and 5 texture-oriented) with high-resolution images and pixel-level annotations. They emulate an industrial inspection setting and have been widely used to benchmark anomaly localization (ALOC) models – MVTec-AD in particular.

The area under the receiver operating characteristic curve (AUROC) [Fawcett, 2006] and area under the per-region overlap (AUPRO) [Bergmann, 2021a] (introduced in Section 2) have been used to evaluate ALOC, but it has been observed that the extreme class imbalance at pixel level inflates the scores produced by these metrics [Rafiei, 2023; Saito, 2015]. As a result, the

performance numbers on MVTec-AD and VisA reported in the literature are converging towards 100% (Fig. 2.1, 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 (FP) (*i.e.* a normal pattern wrongly flagged as anomalous; *cf*. Fig. 2.1, right).

Furthermore, we argue that the VAD literature urges a metric well-suited to its unique characteristic: the positive class (anomalous) 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 anomaly detection (AD) model is to detect *any* type of anomaly. Our work emphasizes on this unsupervised nature of the problem, which – to the best of our knowledge – has not been addressed by existing metrics. We build a performance metric that is *not* conditioned on anomalies – *i.e.* the model behavior measured by the metric does not depend on known anomalous samples, as it is the case in AUPRO, area under the precision-recall (AUPR), instance average precision (IAP), and F_1-max (Section 2).

We present the area under the per-image overlap (AUPIMO) curve (Section 3). It relies on a clear separation of normal and anomalous images for, respectively, validation and evaluation of models. The per-image overlap (PIMO) curve is similar to the receiver operating characteristic (ROC) and per-region overlap (PRO) curves, with the x-axis and y-axis being, respectively, the an FP and a recall measure. However, PIMO's axes are independent because they only depend, respectively, on normal and anomalous images, thus avoiding pixel-level class imbalance issues seen in AUROC and AUPRO. Its strict validation requirement sets a more challenging task inline with the latest advances in the field. It provides means to comprehensively compare models with image-specific evaluation scores, which enables measuring within-dataset performance variance and programatically assessing worst and best-case samples. Furthermore, our work also tackles cross-paper comparison issues – *e.g.* inconsistent downsampling of annotation masks – by proposing the evaluation procedure from our benchmark (Section 4) as a standard.

In summary, this chapter 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 providing a recall measure consistent with AD's unsupervised nature;
2. Per-image recall scoring, enabling the analysis of cross-image performance variance ("how much the performance differs from one image to another") and high-speed execution at high resolution both on CPU and GPU (Section 5).
3. Empirical evidence suggesting that MVTec-AD and VisA datasets have *not* been near-solved and that problem-specific model choice is advisible (Section 5).

The scope of area under the per-image overlap (AUPIMO) targets static image-related applications as in MVTec-AD and VisA. An extension to video-related applications is demonstrated in Section 5.3. Other extensions (*e.g.* 3D images, point clouds, and time series) are out of the scope of this work but we briefly discuss them in Section 6.

Table 2.1: Notation.

Symbol	Description
N, i	Number and index of images
M, j	Number and index of pixels in an image
$\neg, \wedge$	Logical negation, AND
$\lvert\cdot\rvert$	Cardinality of a set or number of 1s in a mask
$\mathbf{a} \in \mathbb{R}_+^M$	Anomaly score map
$\mathbf{y} \in \{0,1\}^M$	Ground truth binary mask
$\mathbf{r} \in \{0,1\}^M$	Region binary mask
$\mathcal{A}$, $\mathcal{Y}$, $\mathcal{R}$	Sets of $\mathbf{a}$, $\mathbf{y}$, and $\mathbf{r}$
$t \in \mathbb{R}_+$	Binarization threshold
$\mathbf{a} \geq t$	Binarization of $\mathbf{a}$ by t(output is in $\{0,1\}^M$)

2 Background

In this section we review metrics commonly used in the VAD literature. We show that most of them employ a validation-evaluation scheme based on threshold choice. This strategy consists of selecting a single or a range of (anomaly score) thresholds based on a validation criterion, and then using a different criterion to compare models. In Section 3 we employ this same scheme to propose AUPIMO while introducing an unbiased validation criterion. Key notation is listed in Table 2.1.

General Framework Our goal is to compare a model's output $\mathbf{a}$ with its ground truth mask $\mathbf{y}$. An anomaly score map $\mathbf{a}$ is a one-channel image where higher values mean "more likely to be anomalous". The range of values in a score map may be restricted to a closed interval, but most AD models output unbounded scores – which is further assumed in this chapter. A ground truth mask $\mathbf{y}$ has 0 and 1 (pixel-level) labels indicating "normal" and "anomalous" respectively. Fig. 2.2b shows an example of $\mathbf{a}$ and $\mathbf{y}$. We define $\mathbf{r}$ as a region in $\mathbf{y}$ such that instances do not overlap (maximally connected components).

We define the false positive rate (FPR) F and the true positive rate (TPR) T (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):

$$\mathrm{F_s} : t \mapsto \frac{\sum_{\mathbf{y}\in\mathcal{Y}} |(\mathbf{a} \geq t) \wedge (\neg\mathbf{y})|}{\sum_{\mathbf{y}\in\mathcal{Y}} |\neg\mathbf{y}|} \qquad (\textit{set-scoped} \text{ FPR (sFPR)}) \qquad , \tag{2.1}$$

$$\mathrm{T_s} : \quad t \mapsto \frac{\sum_{\mathbf{y}\in\mathcal{Y}} |(\mathbf{a} \geq t) \wedge \mathbf{y}|}{\sum_{\mathbf{y}\in\mathcal{Y}} |\mathbf{y}|} \qquad (\textit{set-scoped} \text{ TPR (sTPR)}) \qquad , \tag{2.2}$$

$$\mathrm{F_i} : \quad t \mapsto \frac{|(\mathbf{a} \geq t) \wedge (\neg\mathbf{y})|}{|\neg\mathbf{y}|} \qquad (\textit{image-scoped} \text{ FPR (iFPR)}) \qquad , \tag{2.3}$$

$$\mathrm{T_i} : \quad t \mapsto \frac{|(\mathbf{a} \geq t) \wedge \mathbf{y}|}{|\mathbf{y}|} \qquad (\textit{image-scoped} \text{ TPR (iTPR)}) \quad , \text{ and} \tag{2.4}$$

$$\mathrm{T_r} : \quad t \mapsto \frac{|(\mathbf{a} \geq t) \wedge \mathbf{r}|}{|\mathbf{r}|} \qquad (\textit{region-scoped} \text{ TPR (rTPR)}) \qquad . \tag{2.5}$$

Instances at each scope ($\mathbf{r}$, $\mathbf{y}$, $\mathbf{a}$, $\mathcal{Y}$, $\mathcal{A}$) are ommited in the notation for brevity. All metrics are *pixel-wise* (one score/annotation per pixel), not *image-wise* (one score/annotation per image)

Area Under the Receiver Operating Characteristic Curve (AUROC) The ROC curve (Fig. 2.3a) can be defined as

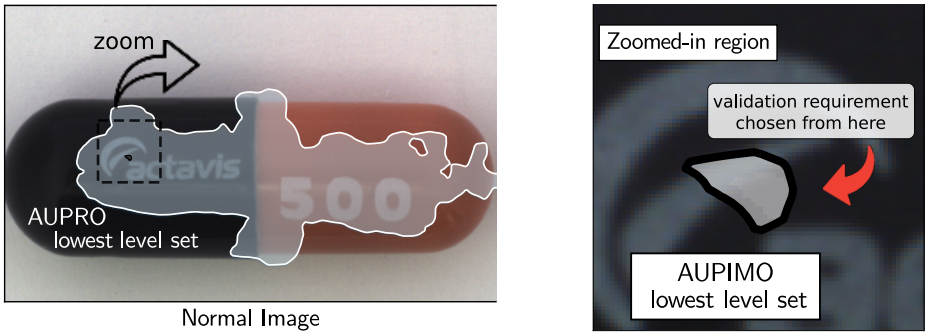
$$\mathrm{ROC} : t \mapsto (\mathrm{F_s}(t)\,,\, \mathrm{T_s}(t)) \quad . \tag{2.6}$$

It traces the trade-off between false positives (FP) and true positives (TP) across all potential binarization thresholds. Its area under the curve (AUC) summarizes the curve into a single score, providing a threshold-independent metric for binary classifiers [Fawcett, 2006]:

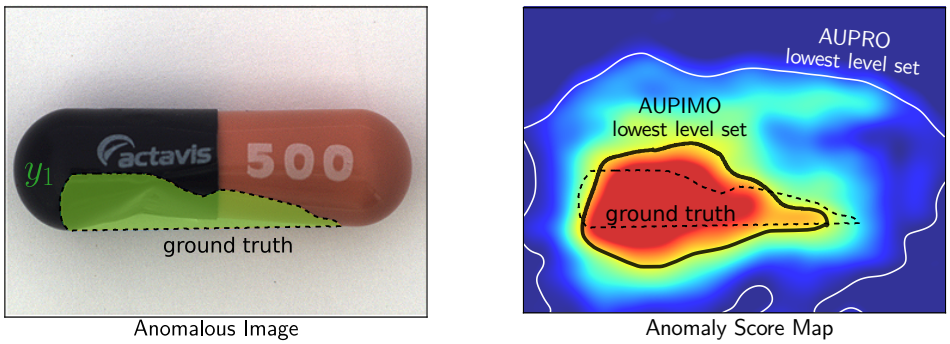
$$\mathrm{AUROC} \;=\; \int_0^1 \mathrm{T_s}\left(\mathrm{F_s^{-1}}(z)\right) \mathrm{d}z \quad , \tag{2.7}$$

where $\mathrm{F_s^{-1}}$ is the inverse of $\mathrm{F_s}$. In practice, it is computed using the trapezoidal rule with a discrete curve given by a sequence of anomaly score thresholds. AUROC is widely used to assess ALOC, treating it as a pixel-level binary classification. However, it has been shown that AUROC is not suitable for ALOC datasets due to the extreme class imbalance [Saito, 2015; Rafiei, 2023]. Besides, it has been argued that, in real-world applications, full or partial localization of anomalous regions is more relevant than pixel accuracy [Zhang, 2023c; Bergmann, 2019]. Finally, being a threshold-independent metric limits its meaning for real-world applications that require threshold selection for inference. These issues prompted the exploration of other evaluation metrics in the field.

Area Under the Per-Region Overlap Curve (AUPRO) Bergmann et al. [Bergmann, 2019] proposed the area under the per-region overlap (AUPRO), which is inspired on AUROC. At each binarization threshold, it measures the region-scoped recall averaged across all anomalous



(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 Fig. 2.5). 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 2.2: 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.

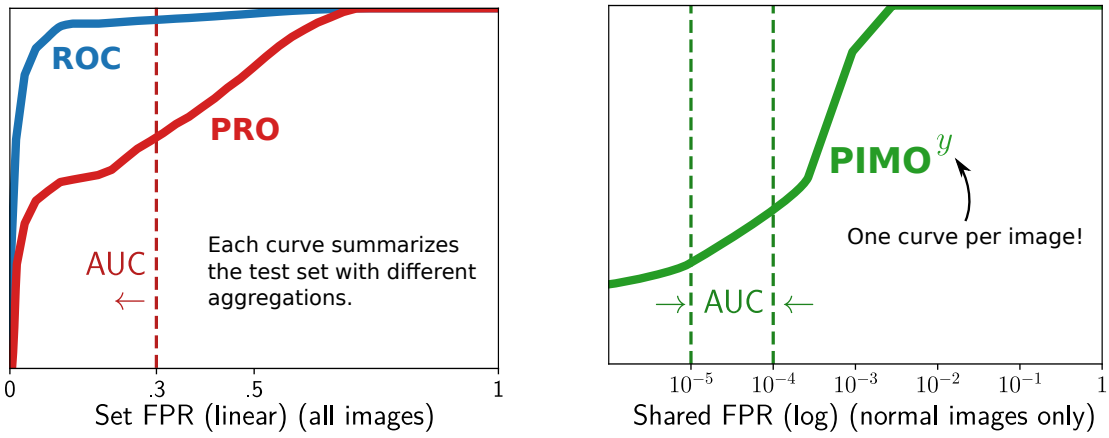
regions available in the test set:

$$\mathrm{PRO} : t \mapsto \left(\mathrm{F_s}(t)\,,\, \overline{\mathrm{T_r}}(t)\right) \quad , \tag{2.8}$$

where

$$\overline{\mathrm{T_r}} : t \mapsto \frac{1}{|\mathcal{R}|} \sum_{\mathbf{r} \in \mathcal{R}} \mathrm{T_r}(t) \tag{2.9}$$

is the average rTPR, and $\mathcal{R}$ contains all $\mathbf{r}$ from all $\mathbf{y} \in \mathcal{Y}$. Like the ROC curve, the PRO curve traces the trade-off between false positives (FP) and true positives (TP) across all potential binarization thresholds. Both use the sFPR of the test set as the x-axis, but different recall measures as the y-axis, reflecting distinct rTPR aggregation strategies. PRO calculates the arithmetic average, giving equal weight to each region. ROC uses the sTPR, which is equivalent to averaging the rTPRs with region size weighting, which makes AUROC more sensitive to large regions.



(a) ROC and PRO curves

(b) PIMO curve

Figure 2.3: 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 $[0, U]$ (where $U = 0.3$ by default[1]), and AUPIMO in $[L, U]$ (where $L = 10^{-5}$ and $U = 10^{-4}$ by default).

The AUC of the PRO curve similarly summarizes it into a single score:

$$\text{AUPRO} = \frac{1}{U} \int_0^U \overline{\text{TP}_\text{r}} \left(\text{FP}_\text{s}^{-1}(z) \right) \, \text{d}z \quad . \tag{2.10}$$

However, AUPRO is restricted to thresholds such that $\text{FP}_\text{s}(t) \in [0, U]$, where U is the upper bound false positive rate (FPR). That restriction corresponds to only using the part of the curve to the left of the vertical line in Fig. 2.3a. Looking at an anomaly score map, it corresponds to considering level sets higher than a certain minimum level set (*cf.* the white level sets in Fig. 2.2). Visually, it corresponds to all contours inside the white contours.

Validation vs. Evaluation With the normalization by the size of the integration range (the term $\frac{1}{U}$), AUPRO effectively computes the average recall (y-axis) from a range of thresholds chosen from an FPR-based requirement. The default value of $U = 30\%$[1] is based on the intuition that, at such sFPR levels, the segmentation contours of the anomalies are no longer meaningful [Bergmann, 2021a]. From this perspective, the FPR restriction in AUPRO acts as a model validation requirement. In other words, the operating thresholds are indexed by the x-axis of the curve (the sFPR), which is used to condition the model behavior to operate over a range of meaningful thresholds. The y-axis (recall) is then used to compare models given that they have been conditioned in the same way. We further refer to this strategy as *validation-evaluation* scheme, which is also used in our proposed metric AUPIMO in Section 3.

Area Under the Precision-Recall Curve (AUPR) Rafiei et al. [Rafiei, 2023] observed that the high pixel-level class imbalance in MVTec-AD and similar ALOC 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). The PR curve has recall as the x-axis and precision as the y-axis, and its AUC is noted AUPR. Similar to the AUCs defined in Eq. (2.7) and Eq. (2.10), it expresses an average of the y-axis. However, unlike ROC and PRO, the PR curve uses the sTPR – instead of the sFPR – to index the operating thresholds. Therefore it provides a precision-based comparison (evaluation) conditioned to recall-achievable results.

Instance Average Precision (IAP) Zhang et al. [Zhang, 2023c] proposed the IAP, a modified version of the PR curve where recall is defined at the region-level. Anomalous regions are considered as detected if at least half of its pixels are correctly detected, and recall is defined as the ratio of the correctly detected regions at each threshold. This alternative recall metric is then used as a validation requirement (threshold choice) and the pixel-level precision is used to compare models (precision-at-k%-recall). While using different metrics, this approach also follows a validation-evaluation scheme, but the threshold is fixed to a single value instead of a range.

F_1-max Other authors have used the F_1 score (the harmonic mean of precision and recall), which strikes a balance between the recall and precision. [Zou, 2022; Jeong, 2023] proposed to use the F_1-max score, which is the maximum F_1 score across all possible thresholds. Otherwise said, it implies a single operating threshold choice. In this case, both the validation and evaluation use the same metric.

All these metrics directly or indirectly condition the model on the anomalous samples present in the test set. IAP and F_1-max directly depend on the anomalous regions to index and select operating thresholds. While AUPR does not restrict the AUC of the PR curve, the thresholds bellow a certain value – where recall is zero – are implicitly discarded. This design creates a bias in favor of detectable anomalies, making the metric sensitive to the distribution of known anomalies. AUROC and AUPRO indirectly account for the anomalies because they account for normal pixels in anomalous images. This contaminates the mapping threshold-FPR because these pixels may contain imprecisions from the annotation process and carry information from neighboring anomalous regions.

[Gangopadhyay, 2022] proposed a benchmarking framework that categorizes images in a dataset-agnostic manner to capture higher-level descriptions of anomalies (*e.g.* structural, surface, or contamination). It enables evaluation across different datasets, providing a more generalized understanding of the models' performance. Their framework introduces a threshold selection based on a validation set. This relates to our validation-evaluation scheme, but they proposed to optimize several metrics depending on anomalous samples. Such validation process

does not address the essential issue of conditioning the model behavior on known anomalies.

3 Area Under the Per-Image Overlap Curve (AUPIMO)

PRO measures region-scoped recall (thus in each individual region) by indexing thresholds with a the set-scoped FPR (the x-axis), which is *shared* by all region instances. We generalize this idea and employ the term *shared* FPR (shFPR) ($\mathrm{F_{sh}}$) to refer to "*any* FP measure" (*i.e.* not necessarily the set set-scoped FPR). The term "shared" reminds that one FP value refers to a threshold that is applied to all anomalous instances, thus is *shared*. In our approach, the sFPR used as x-axis by ROC and PRO is replaced by the average iFPR on normal images only:

$$\mathrm{F_{sh}} : t \mapsto \frac{1}{|\mathcal{Y}^0|} \sum_{\mathbf{y} \in \mathcal{Y}^0} \mathrm{F_i^{\mathbf{y}}}(t) \quad , \tag{2.11}$$

where $\mathcal{Y}^0 \subset \mathcal{Y}$ contains only and all normal images in $\mathcal{Y}$, and $\mathrm{F_i^{\mathbf{y}}}$ refers to the $\mathrm{F_i}$ computed on the instance $\mathbf{y}$. This design choice is a major counterpoint with previous approaches, and its implications are discussed in the following subsections. The **per-image overlap (PIMO)** curve (Fig. 2.3b) is defined as

$$\mathrm{PIMO^{\mathbf{y}}} : \; t \mapsto \left(\log\left(\mathrm{F_{sh}}(t)\right) \, , \, \mathrm{T_i^{\mathbf{y}}}(t)\right) \quad \text{and} \quad , \tag{2.12}$$

and its AUC is, noted AUPIMO, is defined as

$$\mathrm{AUPIMO^{\mathbf{y}}} = \frac{1}{\log\left(U/L\right)} \int_{\log(L)}^{\log(U)} \mathrm{T_i^{\mathbf{y}}}\left(\mathrm{F_{sh}^{-1}}(z)\right) \, \mathrm{d}\log(z) \tag{2.13}$$

$$= \frac{1}{\log\left(U/L\right)} \int_{L}^{U} \mathrm{T_i^{\mathbf{y}}}\left(\mathrm{F_{sh}^{-1}}(z)\right) \, z^{-1} \, \mathrm{d}z \quad , \tag{2.14}$$

where the integration bounds have default values (discussed in Section 3.2) $L = 10^{-5}$ and $U = 10^{-4}$. 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 curve/score per image (superscript $^{\mathbf{y}}$).

The following subsections discuss the implications of the design choices used in AUPIMO that diverge from its predecessors: considering only normal instances in the validation (Section 3.1), using a stricter requirement to condition the model behavior (Section 3.2), evaluating metrics at the image scope, and keeping individual scores for each image (Sections 3.3 and 3.4).

3.1 Bias-free Validation

AUPIMO uses a validation-evaluation strategy similar to AUPRO by restricting the FPR range and computing an average recall over a range of thresholds (indexed by the same metric from

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

the validation). However, 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 should 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.

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. The positive class in AD can have an unlimited number of modes and, in general, no prior information is available about the anomalies. A specific application may, in contrast, have prior information about certain expected types of anomalies (*e.g.* scracthes in an industrial setting as in MVTec-AD's datasets). However, we focus on establishing a fair way to benchmark VAD models across datasets to create a common ground to understand the advances in the field.

Therefore, we adopt the assumption that no prior information about the anomalies is available at training and inference time. As a consequence, we argue that the evaluation process should avoid conditioning the model behavior based on information from anomalous samples because it carries a bias towards *known* anomalies. Specifically, we propose that selecting a single or a range of thresholds at the validation should be based on normal images only. Unseen normal images can be reasonably assumed from the same distribution as the training set, while anomalous images cannot. As such, the x-axis in AUPIMO ($\mathrm{F_{sh}}$) is built only from normal samples.

This essential change avoids biasing the metric towards available anomalies, which is consistent with the unsupervised nature of AD. Note that an alternative version of AUPRO could be defined in the same way, but AUPIMO also carries additional advantages discussed in the following subsections.

3.2 Low Tolerance for False Positives

From an application perspective, anomalies are expected to contain information deserving the user's attention. Yielding too many FPs can lead to user frustration and diminish trust in the model. The validation used in AUPIMO intends to build a more challenging task for VAD models. To achieve this, AUPIMO's x-axis is in log-scale and the integration bounds are set to visually-meaningful values.

The log-scaled x-axis provides a better resolution at low FPR levels and gives more weight to lower FPR levels. Notice how the term $1/z$ in Eq. (2.14) weights the values of the integral giving more importance to the lower z levels since $0 < z < 1$. Besides, the log-scale makes it possible to observe the model's behavior at different orders of magnitude of FPR levels (*cf.* Figs. 2.3b and 2.5).

The integration upper bound U is chosen to yield insignificant FP regions in normal images. Fig. 2.5 shows examples of normal images with FP regions to build an intuition of what this validation constraint represents. Normal images from all the datasets in MVTec-AD and VisA were superposed by FP masks at different iFPR levels. Recall that it corresponds to level sets on an anomaly score map. One normal sample from each dataset is shown with a zoom on

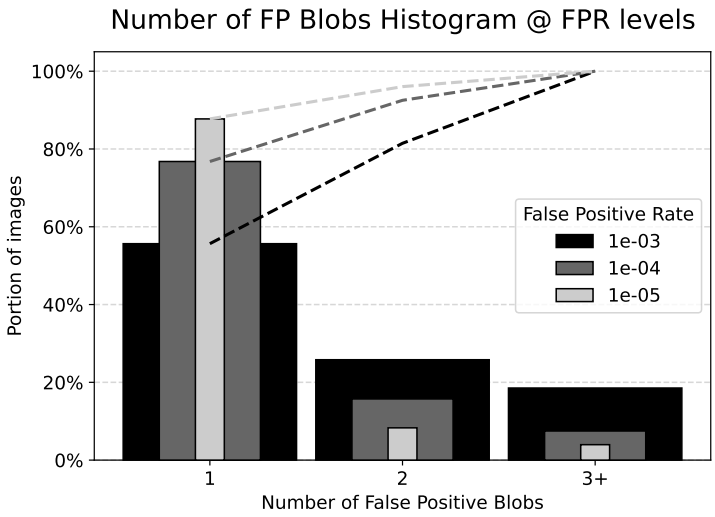


Figure 2.4: False positive regions in normal images. Frequency histograms represented as vertical bars and cumulative histograms as lines. The histograms show the distribution of the number of blobs per image at different iFPR levels. All datasets and all models in our benchmark confounded.

the right (the zoomed area is highlighted in the original image with a dashed rectangle). The masks are generated from anomaly score maps produced by randomly picked models from our benchmark (Section 4) – but FPR levels within each image come from the same map.

Finally, the integration lower bound L needs to be set high enough to avoid numerical issues. Due to the $1/z$ term in the integral, the lower bound needs to be high enough relative to the image size. For most datasets in MVTec-AD and VisA, images have around 10^6 pixels, so $L = 10^{-5}$ is roughly equivalent to 10 pixels. This avoids giving too much weight to the FPR levels close to zero, where discretization effects can have a significant impact.

Notice how, inside AUPIMO's integration bounds ($[10^{-5}, 10^{-4}]$, *i.e.* between black and white in Fig. 2.5), FP regions are barely visible at the image scale and small compared to the structures seen in the images. Fig. 2.4 also shows that, at such low iFPR levels, the number of indendent regions in a normal image tends to be less than three.

In other words, AUPIMO's validation conditions the evaluated model to having very few FPs with insignificant size in normal images. Therefore, an AUPIMO score can be interpreted as the "*average segmentation recall in an image given that the model (nearly) does not yield FP regions in normal images*".

We propose the aforementioned parameters for the datasets from MVTec-AD and VisA as they establish a more challenging task in-line with recent advances in VAD research, but they can be adapted according application-specific needs.

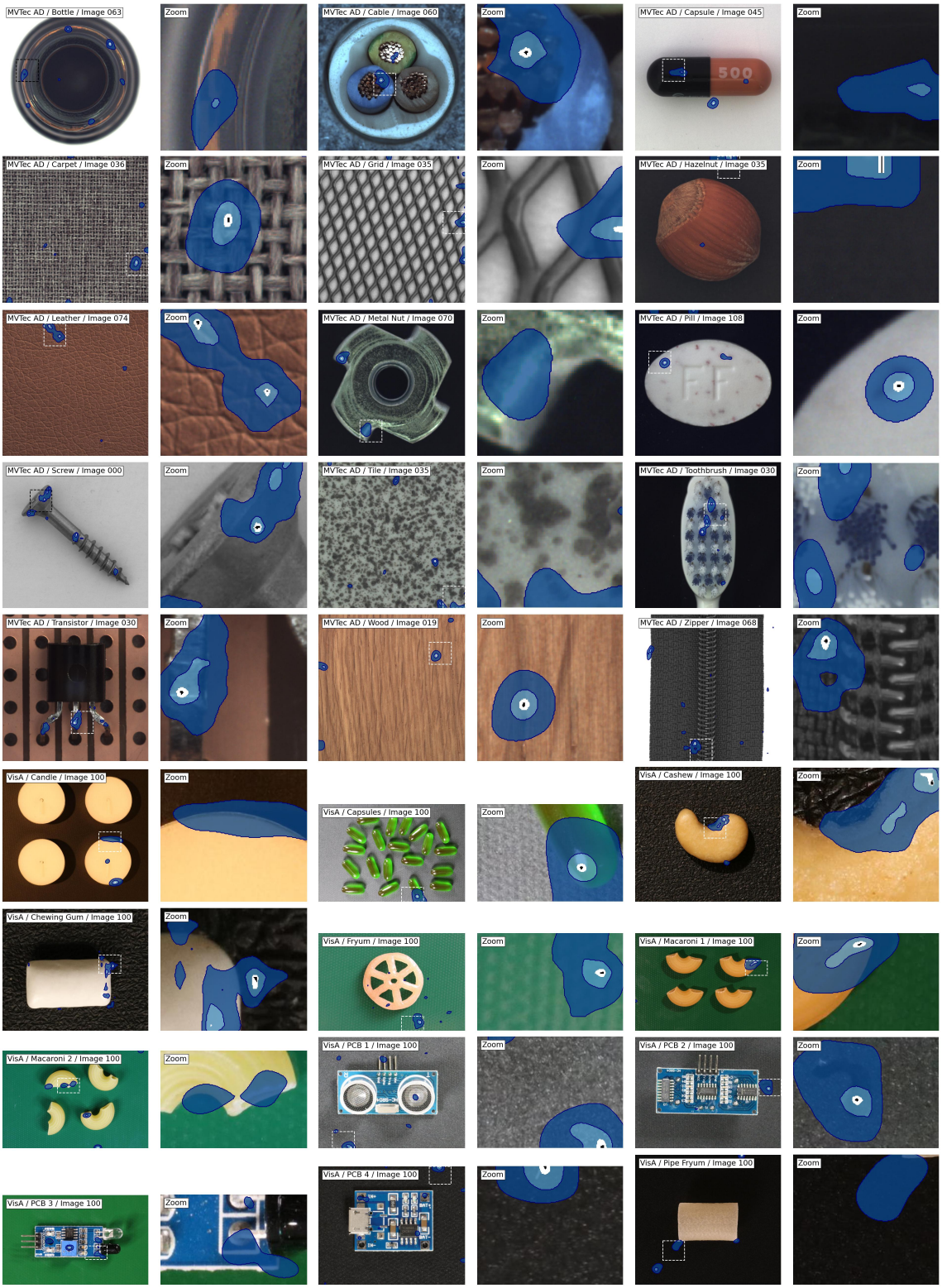


Figure 2.5: Visual intuition of Image False Positive Rate (ImFPR) levels on normal images. Images are normal samples from the datasets in MVTecAD and VisA. Each color corresponds to a predicted mask at a given ImFPR level: darker blue is 10^{-2}, lighter blue is 10^{-3}, white is 10^{-4}, and black is 10^{-5}. The predictions are from the models in the benchmark (randomly selected).

3.3 Image-scoped Evaluation

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), which accounts for image structure and is fast to compute (Fig. 2.6).

A region-based recall metric, as in AUPRO and IAP, would provide a more detailed analysis and be unbiased by the region size. However, computing region-based scores would bring other issues, which are discussed in the following paragraphs.

Image-specific Scores Since each curve/score refers to an input image file, it is easy to index scores to instances. A standard `.json` format is proposed in Listing 2.1 below. The fields `paths` and `aupimos` contain the paths to the input images and their corresponding AUPIMO scores. For benchmark datasets like MVTec-AD and VisA, the paths are truncated to the dataset root directory, making it easy to share results. This format is implemented in our repository[2].

Listing 2.1: Standard `.json` format to publish AUPIMO scores.

```
{
    // the metric used as shared FPR
    "shared_fpr_metric": "mean_perimage_fpr",

    // integraion bounds
    "fpr_lower_bound": 0.00001,
    "fpr_upper_bound": 0.0001,

    // list of paths to input images
    // and their corresponding AUPIMO scores
    "paths": [
        "MVTec/bottle/test/broken_large/000.png",
        "MVTec/bottle/test/broken_large/001.png",
        "MVTec/bottle/test/broken_large/002.png",
    ],
    "aupimos": [0.72107, 0.02415, 0.98991]
}
```

Achieving the same with region-based scores would require more metadata to ensure a one-to-one correspondence between instances (regions in that case) and scores. Such approach would require finding maximally connected regions in the annotations, which can be implementation-sensitive. For instance, Anomalib's [Akcay, 2022] CPU and GPU-based implementations are respectively from `opencv-python` [Bradski, 2000] and `kornia` [Riba, 2020], and their AUPRO scores slightly differ. While avoiding these issues, image-specific scores enable fine-grained analyses otherwise impossible with AUROC and AUPRO. Score distributions (*e.g.* Fig. 2.11) –

[2] https://github.com/jpcbertoldo/aupimo

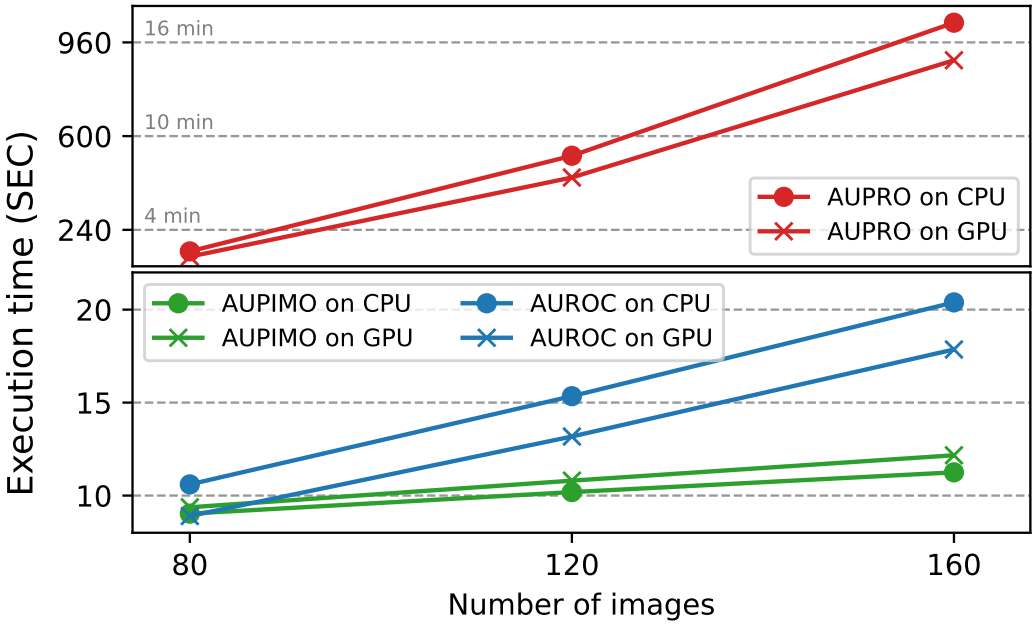


Figure 2.6: Execution time of AUROC, AUPRO, and AUPIMO on MVTec AD / Screw dataset (image resolution of 1024×1024; average times over 3 runs).

instead of single-valued scores – provide insight into performance variance and worst/best case scenarios, which we exploit to select representative samples for qualitative analysis. Finally, it also enables the use of statistical tests, which we showcase in an ablation study in Section 5.3.

Execution Time Having computationally efficient metrics is essential to enable fast iterations and not create computational bottlenecks in research and development. We tested the execution time on the MVTec AD / Screw dataset at full resolution (1024×1024) on CPU and on a single GPU. 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.

Fig. 2.6 shows that AUPIMO is orders of magnitude faster than AUPRO and generally faster than AUROC, both on CPU and GPU. While AUPRO's computational cost is reasonable at downsampled resolution (not covered in Fig. 2.6), at full resolution it becomes impractical – especially on CPU. As argued in Section 4, evaluation at full resolution is highly desirable to fairly compare models across papers.

AUPRO's slow execution is mainly due to the computation of connected component analysis, while AUROC and AUPIMO do not need it. AUPIMO's implementation relies on simple operations, enabling the use of `numba` [Lam, 2015] to further accelerate the computation (reported execution times include the just-in-time compilation).

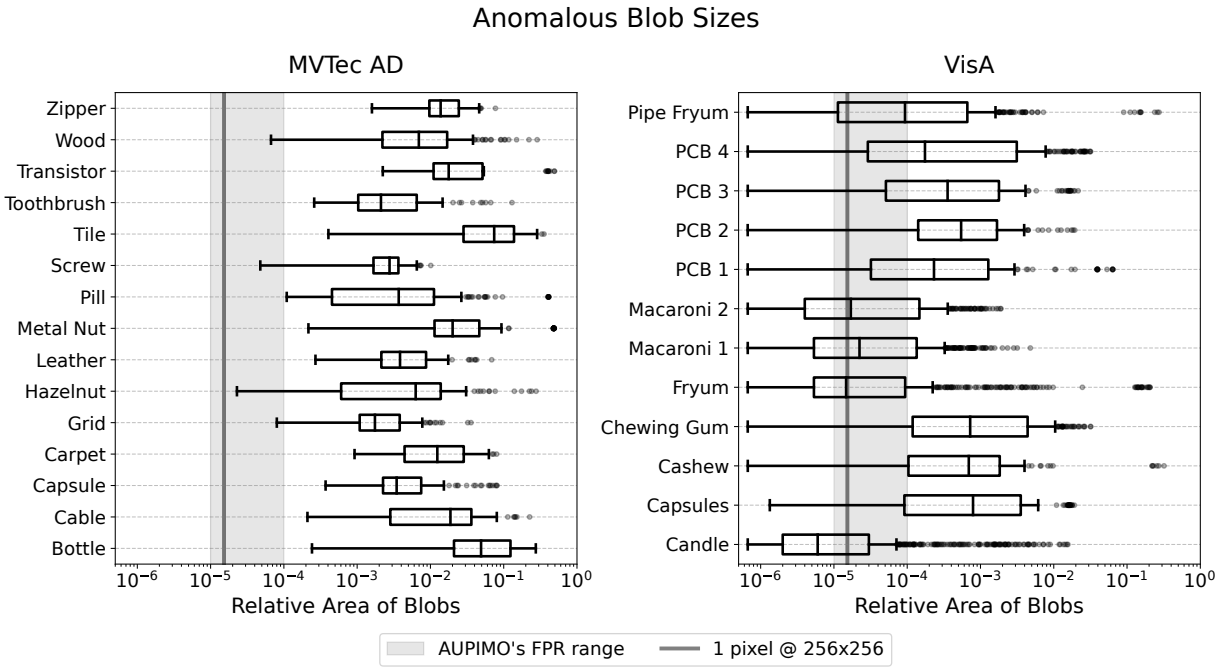


Figure 2.7: Distribuition of relative size of anomalous regions (from the test set annotations) in the datasets from MVTecAD and VisA.

3.4 Robustness to Noisy Annotation

In real-world use-cases, high-quality annotation is hard to acquire or even to define. As a result, the evaluation process can be contaminated by noisy annotations. We found that the annotations in the VisA dataset contain noisy, meaningless regions. In this section, we show evidence that this issue is prevalent in the VisA and that AUPIMO is more robust to this issue than AUPRO.

Fig. 2.7 shows the distributions of the relative region size in ground truth annotations in each dataset from MVTec-AD and VisA. Relative size is the number of pixels in a maximally connected component divided by the number of pixels in the image. Lower and upper whiskers are set with maximum size to 1.5 inter-quartile range (IQR), and fliers (outliers) are shown as gray dots. The gray-shaded span is AUPIMO's integration range.

The vertical gray line represents the relative size of a single pixel at resolution 256×256, which is a common input resolution in the literature and in our experiments. This provides a reference value to understand the distributions: anomalous regions with relative size $\leq \frac{1}{256^2}$ are not visible to the models.

The size of the anomalies in MVTec-AD are generally between 10^{-3} and 10^{-1} (few cases are as small as 10^{-4}). Since (almost) none of the anomalies are as small as the FPR integration range proposed in Section 3.1, AUPIMO scores can be interpreted as a (near) FP-free recall measure. In practical applications, one could leverage this validation to dismiss any prediction with relative size below the integration range.

The anomalies in VisA are generally smaller than those from MVTec-AD, and the distribution has a long queue of small anomalies. In most dataset, the majority of anomalies are larger than AUPIMO's upper bound, but some of them are concentrated inside its bounds. All

Table 2.2: Statistics from tiny blobs in VisA [Zou, 2022].

(a) Tiny region sizes (absolute). The numbers in the table are counts of regions.

Reg Size (abs) Category	1 - 9	10 - 19	20 - 29
Candle	358	98	20
Capsules	8	7	3
Cashew	10	0	1
Chewing Gum	39	1	0
Fryum	158	96	22
Macaroni 1	114	52	14
Macaroni 2	123	54	6
PCB 1	19	20	9
PCB 2	11	8	4
PCB 3	20	11	0
PCB 4	32	19	12
Pipe Fryum	44	34	9

(b) Number of regions per image. The numbers in the table are counts of images.

Nb Reg/Img Category	1 - 5	6+	Total
Candle	5	18	23
Capsules	6	1	7
Cashew	5	0	5
Chewing Gum	6	1	7
Fryum	22	13	35
Macaroni 1	27	10	37
Macaroni 2	21	8	29
PCB 1	10	2	12
PCB 2	10	1	11
PCB 3	8	1	9
PCB 4	10	5	15
Pipe Fryum	17	3	20

datasets are skewed to the left (small) showing relative sizes as small as 10^{-6} to 10^{-5}. This range of sizes roughly corresponds to a few tens of pixels in the original resolution, and they are often neglectible in the context of the image. We further refer to regions of size smaller than $\frac{1}{256^2}$ as *tiny regions*.

The examples of tiny regions in Fig. 2.9 show that they are mostly meaningless. They are often near the surroundings of an actual anomaly (e.g. Fryum/Image 048), but there are cases where tiny regions are isolated in a normal zone (e.g. Chewing Gum/Image 068 and Macaroni 2/Image 067). Table 2.2 shows statistics about the absolute sizes (at original resolution) and the number of tiny regions per image in each dataset from VisA. In fact, they are so numerous that the actual anomalous regions show as outliers in Fig. 2.7.

These analyses show evidence that VisA is heavily contaminated by noisy annotation. Fortunately, we found that AUPIMO is robust to this issue.

Robustness 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. 2.8 demonstrates this in an experiment with artificially added noise. Random mistakes mimicking statistics from VisA (Table 2.2) are added to the datasets in MVTec-AD. Histograms in Fig. 2.8 show the distribution of the difference between the scores without and with the synthetic mistakes. Distributions closer to zero mean the metric is more robust to the noise. The results in Fig. 2.8 are from all datasets and 10 seeds all counfounded.

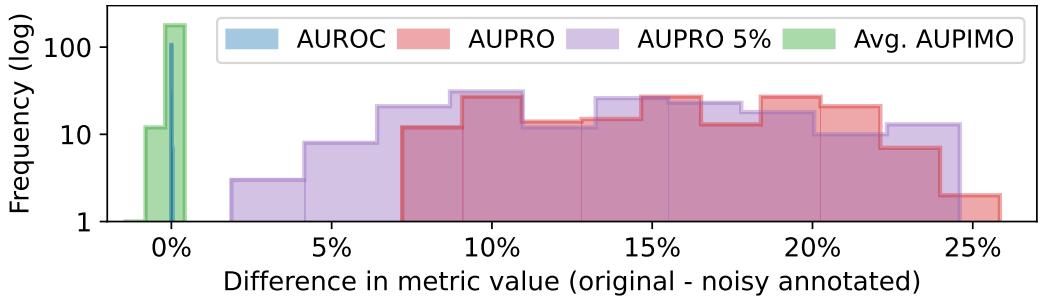


Figure 2.8: Robustness of AUROC, AUPRO, and AUPIMO to noisy annotations randomly added to the MVTec-AD's datasets (score difference histograms; closer to zero is better).

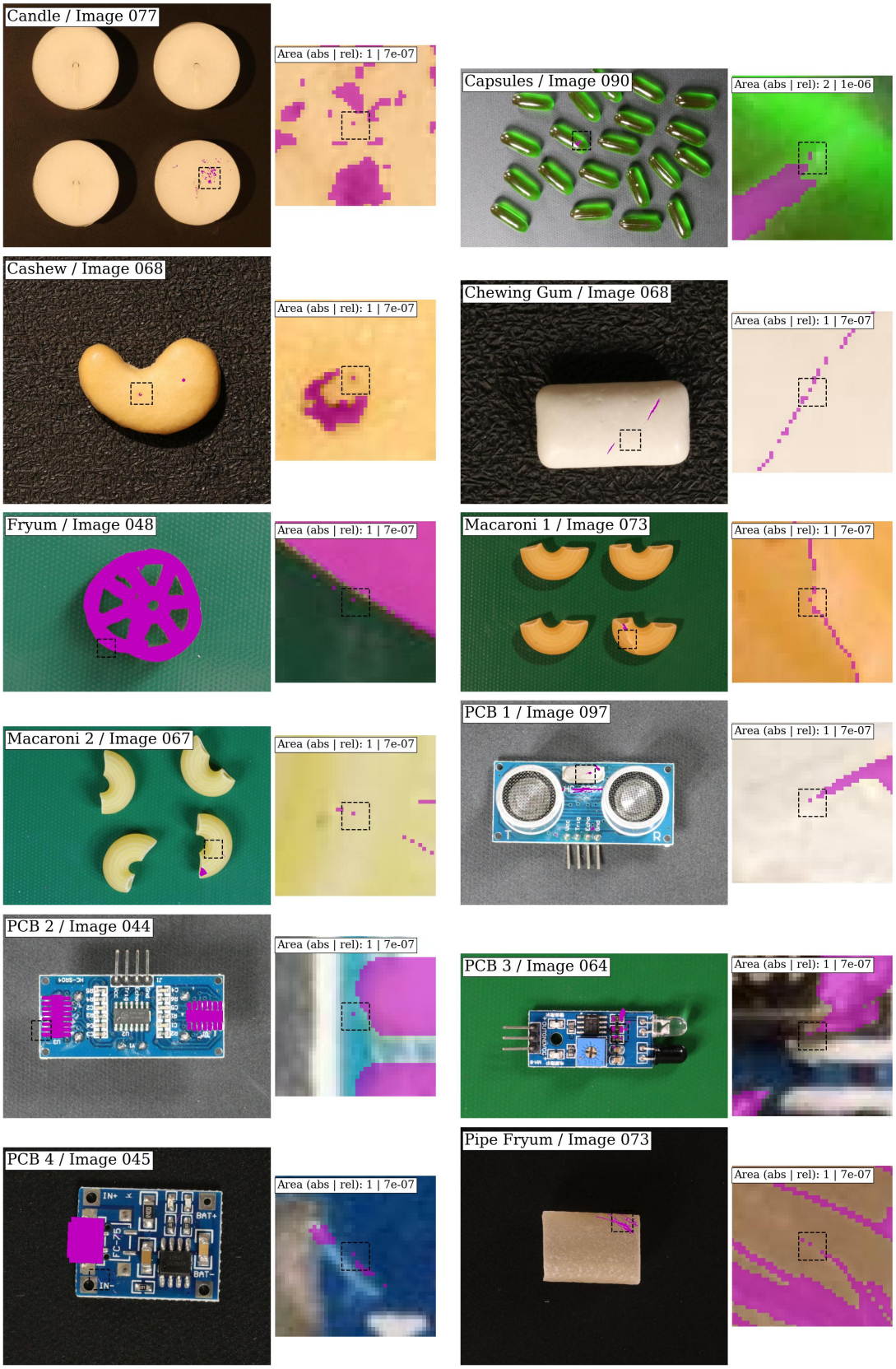


Figure 2.9: Tiny anomalous regions in VisA.

4 Benchmark

We benchmarked the datasets from MVTec-AD (examples in Fig. 1.1) and VisA with state-of-the-art (SOTA) models to compare the performances reported in terms of AUROC, AUPRO, and AUPIMO. We also evaluated AUPRO with $U = 5\%$ ($\text{AUPRO}_{5\%}$) for the sake of comparing with a more challenging alternative of that metric.

Models We reproduced[3] a selection of 13 models from eight papers summarized in Table 2.3. For some models we considered two backbones and selected the (generally) best out of the two to show in some analyses. When the backbone is not specified, the default backbone (see Table 2.3) is the one shown in the results. 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).

We used the following implementations:

- `anomalib` [4] [Akcay, 2022] for PaDiM [Defard, 2021], PatchCore [Roth, 2022], and FastFlow [Yu, 2021];
- github.com/gasharper/PyramidFlow for PyramidFlow [Lei, 2023];
- github.com/donaldrr/simplenet for SimpleNet [Liu, 2023];
- github.com/tientrandinh/revisiting-reverse-distillation for RevDist++ [Tien, 2023];
- github.com/mtailanian/uflow for UFlow [Tailanian, 2024];
- github.com/nelson1425/EfficientAD for EfficientAD [Batzner, 2024].

All models were trained with 256×256 images (downsampled with bilinear interpolation, no center crop), and with the hyperparameters reported in the original papers. The implementations of AUROC and AUPRO are from Anomalib [Akcay, 2022].

Evaluation Guidelines Cross-paper comparisons in the VAD 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; (4) (ideally) report the score distribution (*e.g.* boxplots as in Fig. 2.11). The next paragraphs discuss these guidelines.

[3] 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 [Bergmann, 2021a] `https://github.com/gasharper/PyramidFlow` (commit `6977d5a`), see function `compute_pro_score_fast` in the file `util.py`.

[4] `github.com/openvinotoolkit/anomalib` commit `09ad1d4b1e8f634b72f788314275d3aea33815dd`.

Model	Publication	Backbone	Family	Default	Implem.
PaDiM	ICPR 21	ResNet18	probability density	✓	anomalib
PaDiM	ICPR 21	WideResNet50	probability density	–	anomalib
PatchCore	CVPR 22	WideResNet50	memory bank	–	anomalib
PatchCore	CVPR 22	WideResNet101	memory bank	✓	anomalib
SimpleNet	CVPR 23	WideResNet50	reconstruction	✓	official
PyramidFlow	CVPR 23	ResNet18	normalizing flow	–	official
PyramidFlow	CVPR 23	–	normalizing flow	✓	official
RevDist++	CVPR 23	WideResNet50	student-teacher	✓	official
FastFlow	arXiv (21)	WideResNet50	normalizing flow	–	anomalib
FastFlow	arXiv (21)	Cait M48	normalizing flow	✓	anomalib
EfficientAD-S	arXiv (23)	WideResNet101	student-teacher	–	unofficial
EfficientAD-M	arXiv (23)	WideResNet101	student-teacher	✓	unofficial
UFlow	arXiv (23)	–	normalizing flow	✓	official

Table 2.3: Models in the benchmark. Years were abbreviated to the last two digits. When two backbones are listed, the default is the one shown in the results by default.

Full Resolution Many models typically downsample input images, which conveniently reduces computational costs. However, for a fair and model agnostic evaluation, it is important to use the original resolution as it impacts the decision-making when choosing the most suitable model for a use-case. If a small anomaly is missed due to downsampling, it is desirable to penalize this, while rewarding models that can handle higher resolution. As [Zhang, 2023c] pointed out, downsampling ground truth masks creates artifacts, leading to inconsistent results across papers. When a model's output has lower resolution than the annotation, our recommendation is to apply bilinear interpolation to upsample the anomaly score maps.

No Crop Center crop has been used to leverage the center alignment of the objects depicted in MVTec-AD and VisA. However, this is a prior knowledge, hence we do not apply crop.

Score Publication We recommend to publish AUPIMO scores for all images. A standard format is specified in Section 3.3. All the scores from our benchmark are available in this format in our repository `https://github.com/jpcbertoldo/aupimo`. It is advised to report score distributions (*e.g.* as boxplots and histograms) when possible for a more comprehensive evaluation.

Sample Selection To avoid biases from cherry-picking qualitative samples, we propose a systematic selection procedure. Select the images whose AUPIMO are closest to the statistics in a boxplot: mean, first/second/third quartiles, and low/high whiskers set with maximum size of 1.5 IQR (inter-quartile range). We applied this procedure to select the samples shown in Fig. 2.10. Note that this is not restricted to AUPIMO as it is applicable to any per-instance score (*e.g.* the image-scoped intersection over union (IOU) presented in Chapter 3).

Statistical Tests When applicable, we recommend using statistical tests to assess the significance of the differences between models. It can be applied to within-dataset comparisons; for instance, it can be used to assess the consistency of the performance gain given by different

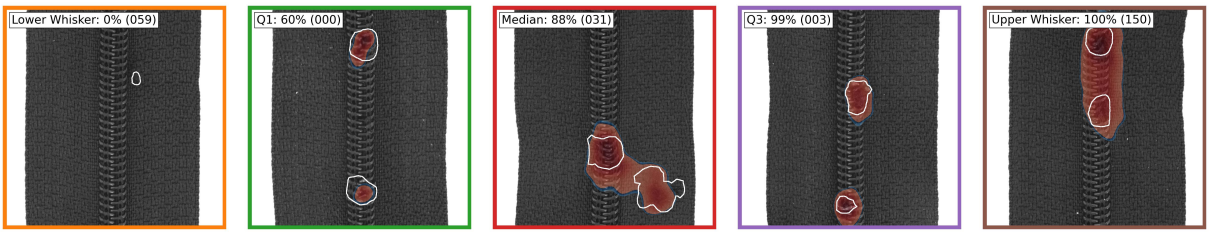


Figure 2.10: Heatmaps on MVTec AD / Zipper. Images selected according to AUPIMO's statistics from the boxplot of the model SimpleNet in Fig. 2.11. Statistic and image index annotated on the upper left corner of each image.

components of the model. We show an example of that use case in Section 5.3. The Wilcoxon signed-rank test [Benavoli, 2016; Demšar, 2006] is used for a non-parametric comparison of pairs of models. The null hypothesis H_0 is that two models A and B are equivalent, thus their average ranks tend to equal. The alternative hypothesis H_1 is that one of the two models (say, A) is *more often* better than B. No assumption is made about the scores distributions making it robust to outliers [Benavoli, 2016; Demšar, 2006]. The term C refers to the confidence to reject the null hypothesis; higher means more confidence on the superiority of A over B. In other words, high confidence ($C = 1 -$ p value) to reject the null hypothesis (*i.e.* low p value) means that A *consistently* outperforms B.

5 Results

In this section, we present the results of our benchmark on the datasets from MVTec-AD and VisA in terms of AUROC, AUPRO, and AUPIMO. First, the results of a showcase dataset (Zipper from MVTec-AD) are presented in detail. For the sake of space, the results of the other datasets are documented in Appendix 1 (extracted from [Bertoldo, 2024]). Then, a discussion about the SOTA in VAD is developed based on an analysis of performance across datasets. Finally, we present extra results to further illustrate the advantages of AUPIMO over AUROC and AUPRO.

5.1 Detailed Example: the Zipper Dataset from MVTec-AD

In this subsection, we provide a detailed analysis of the Zipper dataset from MVTec-AD to illustrate how AUPIMO can reveal more information about the models' performance than its predecessors, AUROC and AUPRO. Fig. 2.11 illustrates two common observations in our benchmarks.

First, it shows how AUROC and AUPRO fail to reveal differences between models with tangible figures. The differences of AUROC and AUPRO between the two best models are, respectively, 0.1% and 0.4%. 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 even the best models have

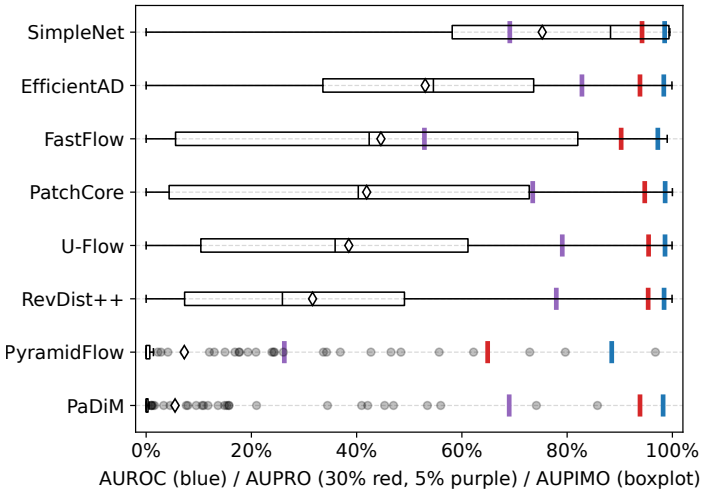


Figure 2.11: Benchmark on dataset MVTecAD / Zipper: AUROC, AUPRO, and AUPIMO scores.

left-skewed AUPIMO distributions – *cf.* the best models per dataset in Fig. 2.18. In Fig. 2.11, several models have worst and best-case samples at 0% and 100% AUPIMO respectively. Fortunately, AUPIMO provides the means to investigate this by programatically identifing specific instances or anomaly types not well-detected by a model, such as the sample on the leftmost side in Fig. 2.10.

Fig. 2.12 shows an alternative analysis. For each image, all models are ranked from 1 (best) to 13 (worst) – "which model best detects this specific image?" – and the average rank is computed for each model. An average rank of 1 would mean that a model has the highest segmentation recall in all images of the dataset. Conversely, an average rank of 13 would mean that a model has the lowest segmentation recall in all images. This non-parametric analysis reveals if a model is consistently better than others across the dataset disregarding the magnitude of the differences.

Fig. 2.12 also shows the confidence on the alternative hypothesis from the Wilcoxon signed-rank test for adjacent models (cliques above the rank scale). When the confidence is higher than 0.95 (*i.e.* p value lower than 0.05), we consider the difference statistically significant and do not show the confidence. See the paragraph about statistical tests in Section 4 for more details.

Fig. 2.12 reveals that the best two models are consistently better than their competitors, but from the third to the eighth models the differences are not statistically significant. These nuances are otherwise impossible to assess with single-valued metrics like AUROC and AUPRO.

5.2 Cross-dataset Analysis

In this subsection, we analyze the models' summarized performance across all datasets from MVTec-AD and VisA.

In Figs. 2.13 to 2.17, each triangle is a single-valued score (AUROC, AUPRO, and $\text{AUPRO}_{5\%}$) or a cross-image statistic (average AUPIMO or average AUPIMO rank) from one model trained in one dataset from MVTec-AD ($\triangle$) or VisA ($\triangledown$). Diamonds ($\diamond$) are cross-dataset (all confounded) or cross-model averages. Notice that plots have different x-axis scales. Fig. 2.13 shows

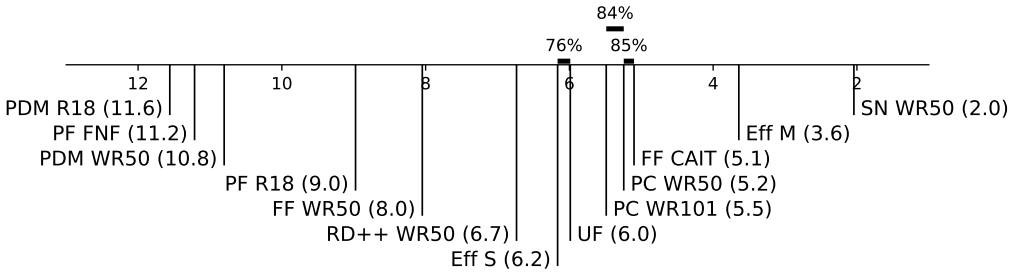


Figure 2.12: Benchmark on dataset MVTecAD / Zipper: AUPIMO average rank diagram. Cliques between two adjacent models show the confidence on the alternative hypothesis from the Wilcoxon signed-rank test when it is lower than 0.95 (*i.e.* p value higher than 0.05). See the paragraph about statistical tests in Section 4 for more details.

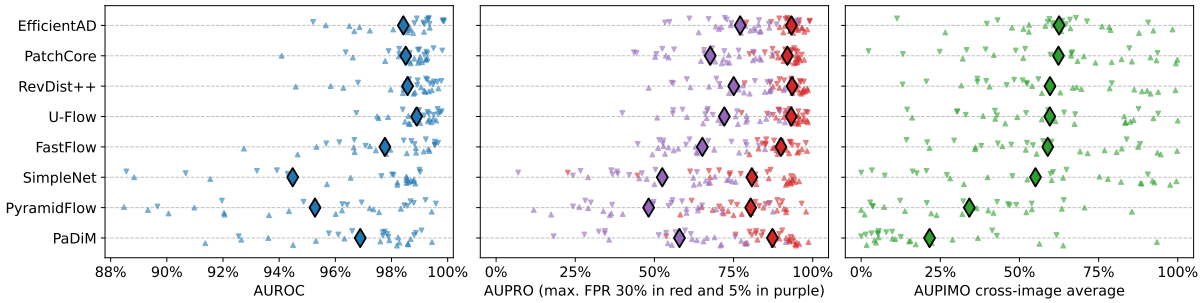


Figure 2.13: Dataset-wise comparison summarized.

a summary with the default backbones only (*cf.* Table 2.3). Figs. 2.14 to 2.17 show all the models with both backbones when applicable. Figs. 2.14 to 2.16 include comparisons grouped by dataset and model. Fig. 2.17 shows the average image ranks according to AUPIMO for each model (computed as in Fig. 2.12). The numerical values of the cross-dataset averages are summarized in Table A.3.

The results from our benchmark reveal two key insights regarding the SOTA in VAD. First, the benchmark datasets from MVTec-AD and VisA still have room for improvement. This major observation contradicts the perception from previous works based on AUROC and AUPRO scores. While $\text{AUPRO}_{5\%}$'s (purple) stricter FPR bound shows to be a 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 have a more practical and trustworthy task to achieve: 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. 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.

Looking at the cross-dataset averages, two groups of models stand out:

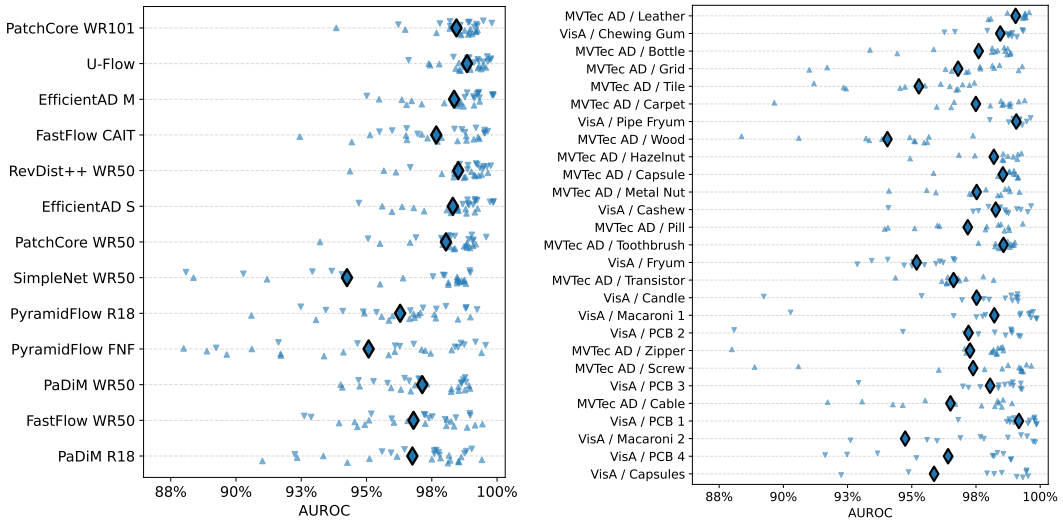


Figure 2.14: Dataset-wise comparison with AUROC.

- **inferior**: PaDiM, FastFlow, and PyramidFlow;
- **superior**: PatchCore, RevDist++, EfficientAD, and UFlow.

Although all the scores show this grouping, the performance gaps in terms of AUROC and AUPRO are numerically small and hard-to-grasp. The differences in terms of AUPIMO are more pronounced. AUPIMO's strict validation makes it clearer when models actually excel in a given dataset, or if it fails under a low FP tolerance. Note that SimpleNet seems to be in the inferior group according to AUROC and AUPRO, but it is in the superior group according to AUPIMO and average rank.

The average rank analysis (Fig. 2.17) further confirms the fact that succeciding in a given dataset is highly model-dependent. For instance, both UFlow, EfficientAD, PatchCore, and SimpleNet have cases with average rank as high as two and cases with average rank as low as nine.

Finally, in Figs. 2.14 to 2.16, the cross-model average scores provide an empirical reference to compare the difficulty of the datasets. In AUROC terms, the averages fluctuate between 96% and 99%, giving the impression that the datasets are overall equally challenging. AUPRO shows more pronounced differences, with averages between 80% and 95%. Finally, AUPIMO's strict validation causes some datasets to become so hard that almost all models fail to achieve a reasonable recall.

Fig. 2.18 shows the AUPIMO distributions (as boxplots) from al datasets. Fig. 2.18b shows the best model per dataset (in terms of average AUPIMO rank), and Fig. 2.18a shows the performances from PatchCore WR101 only (ranked first according to the same criterion). This analysis further sheds light on the current state of the art showing that current models still are not capable of cracking the datasets from MVTec-AD and VisA.

Although PatchCore WR101 is the overall best, it still has a long tail of low performance on several datasets like Grid and Wood – which are rather easy according to Fig. 2.16 (right). In some cases, PatchCore WR101 even practically fails to detect any anomaly at all, like in

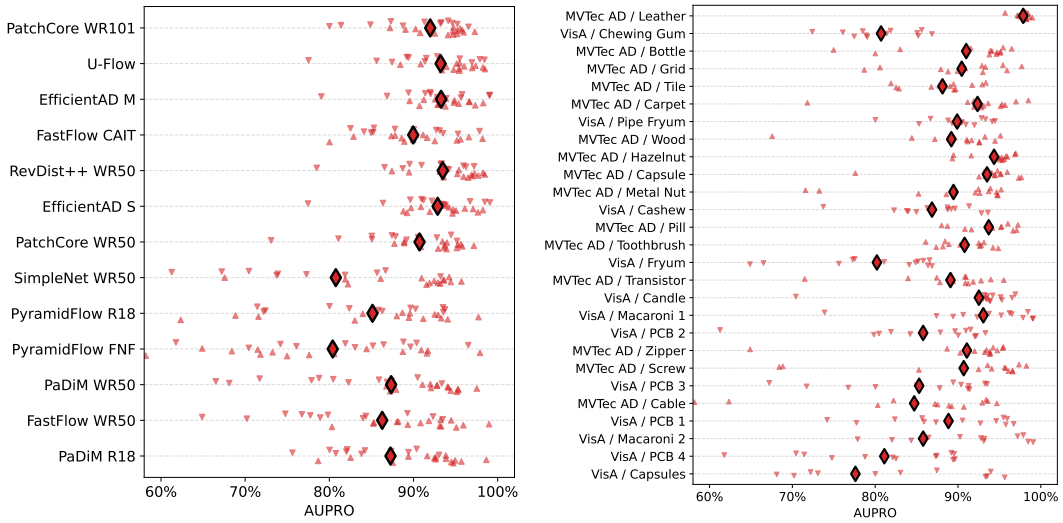


Figure 2.15: Dataset-wise comparison with AUPRO.

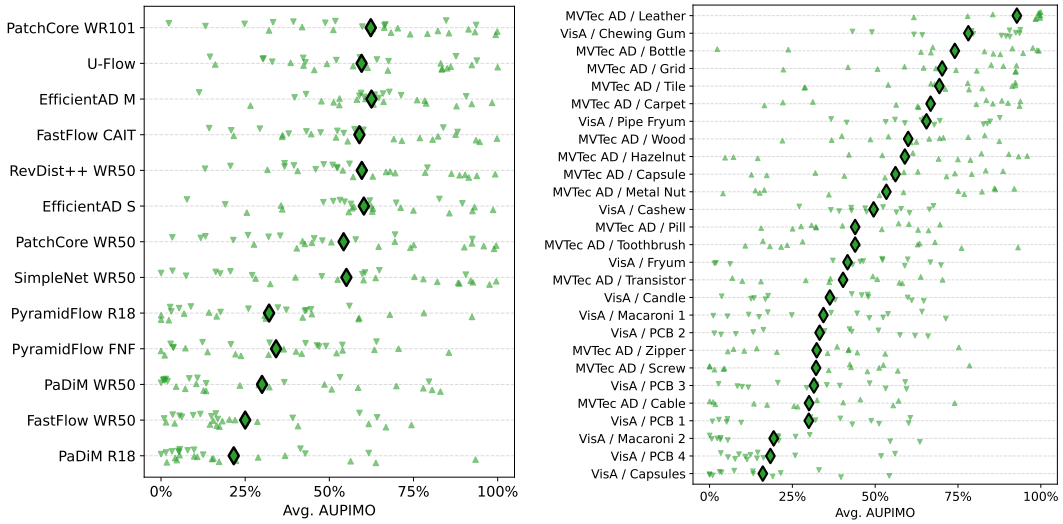


Figure 2.16: Dataset-wise comparison with Avg. AUPIMO.

Capsules and Macaroni 2. Finally, Fig. 2.18b shows that even selecting the best model per dataset would still result in many missed anomalies in the most challenging datasets.

5.3 Other Results

Ablation Study Table 2.4 showcases the use of statistical tests in an ablation study of EfficientAD [Batzner, 2024] on the dataset MVTec-AD / Capsule. The model configurations and the layout of the table were based on Tab. 4 in [Batzner, 2024]. At each row a component is added to the model above, starting with Patch Description Network (PDN) at top and resulting in EfficientAD at the bottom. A Wilcoxon signed-rank test is performed between the model above and the model below (*cf.* paragraph about statistical tests in Section 4). Higher values of C mean more confidence on that the improvement by adding the new component is consistent. Each component generally shows significant improvements, but the bottom right cell is an

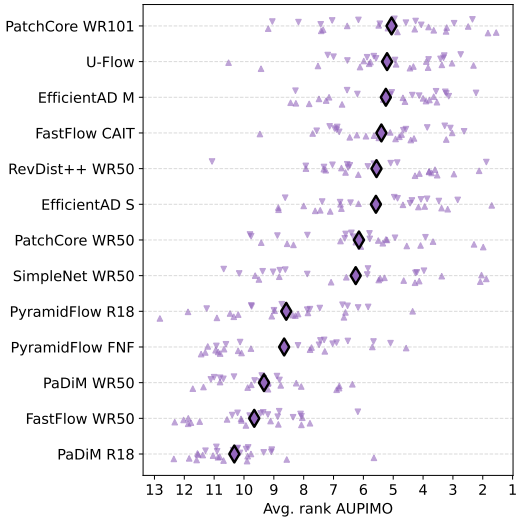


Figure 2.17: Dataset-wise comparison with average rank (according to AUPIMO).

exception. Pretraining penalty causes a score drop, and the low confidence on the alternative hypothesis confirms that the drop is consistent across images.

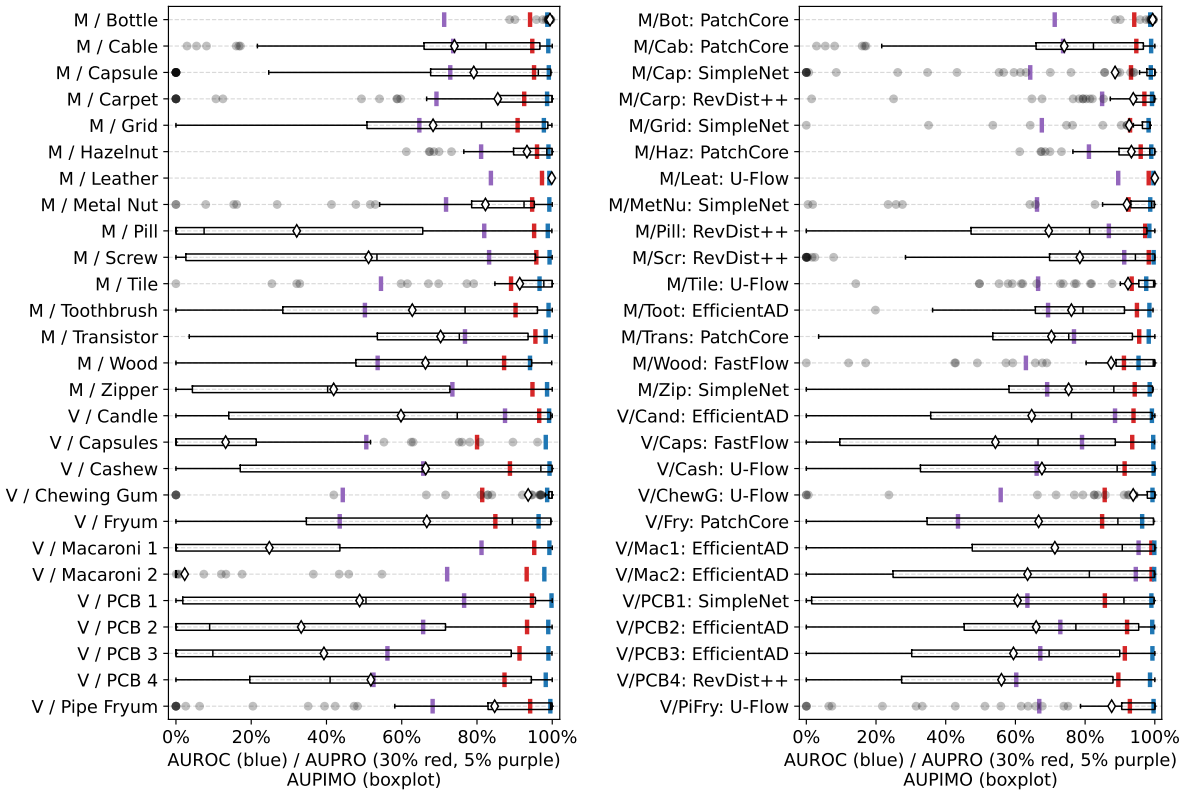
Table 2.4: Ablation study (use-case of statistical tests with AUPIMO).

Avg. AUPIMO (Diff. ↑ [%]; Confidence C ↑ [%])	EfficientAD-S	EfficientAD-M
PDN	~ 0	~ 0
↪ map normalization	22 (+22; 100)	23 (+23; 100)
↪ hard feature loss	57 (+35; 100)	59 (+36; 100)
↪ pretraining penalty	64 (+7; 100)	57 (-2; 0.02)

Does AUPIMO Correlate with AUROC and AUPRO? Fig. 2.19 shows scatter plots of AUROC and AUPRO vs. (cross-image) average AUPIMO. All models and datasets in the benchmark confounded. Notice that the scales of the axes are different for each metric. Both plots seem to show a positive correlation (not as clear between AUROC and AUPIMO), which makes sense because they all measure recall. However, one metric is not enough to imply the other.

High levels of AUROC and AUPRO do not guarantee high levels of AUPIMO. For instance, in the Zipper dataset (Fig. 2.11), PatchCore has the highest AUROC (98.6%) but its average AUPIMO is 41.5% (ranked 5th). Similarly, UFlow has the highest AUPRO (95.5%) but its average AUPIMO is only 38.5% (ranked 6th). Conversely, high levels of AUPIMO *tend* to imply higher levels of AUPRO and AUROC – notice the slightly triangular shape of the point clouds. It is worth to note that all these three models miss some anomalies entirely, like the leftmost sample in Fig. 2.10.

Video We show an example of how AUPIMO can be extended to video applications. It must be stressed that this is not a full-fledged application, but rather a proof of concept. The UCSD



(a) PatchCore-WR101

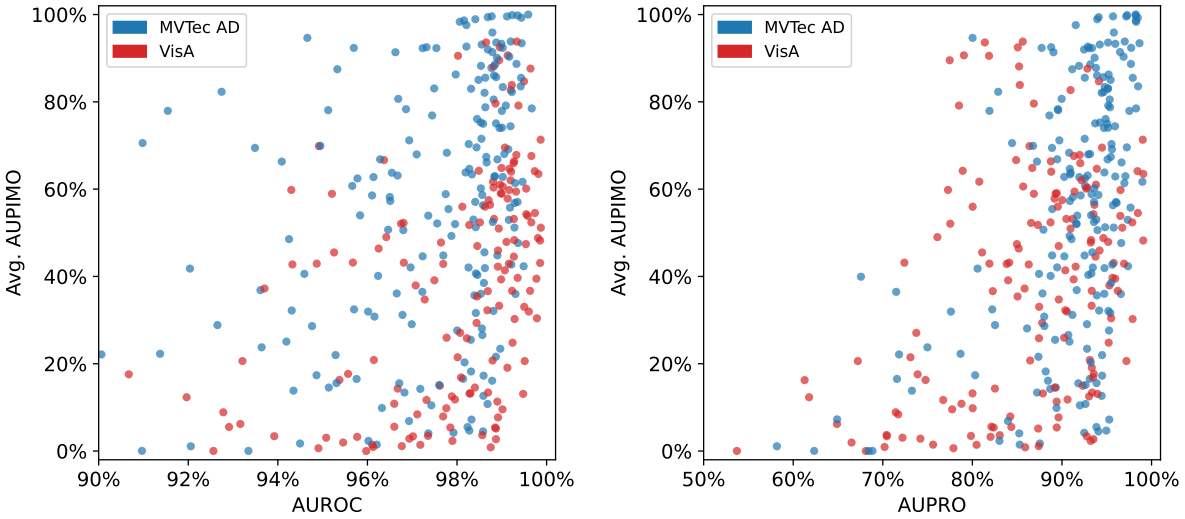
(b) Per-dataset best models

Figure 2.18: AUPIMO distributions for PatchCore-WR101 (left) and per-dataset best models (right).

Pedestrian dataset [Mahadevan, 2010] was used to illustrate this concept because it has been widely used and cited in the literature, but other datasets like A Day on Campus (ADOC) [Pranav, 2020] and Street Scene [Ramachandra, 2020] would also be relevant to this discussion.

A PatchCore [Roth, 2022] model was trained on the UCSD Pedestrian dataset at every two frames. The model was evaluated with the same procedure than our experiments by ignoring the temporal dimension of the videos and treating all the frames from all the videos as a single dataset. Fig. 2.20 shows the AUPIMO scores for each frame in the test videos along the time axis (referenced by the frame index).

A selection of frames from the video Test006 are shown in Fig. 2.21. Notice how AUPIMO's validation works in practice: the normal frame (175) does not have any visible FP region. Frame 61 shows a case where the image-scoped approach has limitations. The AUPIMO score is around 50% because there are two indendent anomalous regions in the frame; one of them is well detected by the model, but the other is ignored. A better modelization for this case would require a more complete annotation where each instance of anomaly is labeled separately, which is not the case in the UCSD Pedestrian dataset.



(a) AUPIMO vs. AUROC

(b) AUPIMO vs. AUPRO

Figure 2.19: Scatter plots of Average AUPIMO vs. {AUROC, AUPRO} on the MVTec AD and VisA datasets.

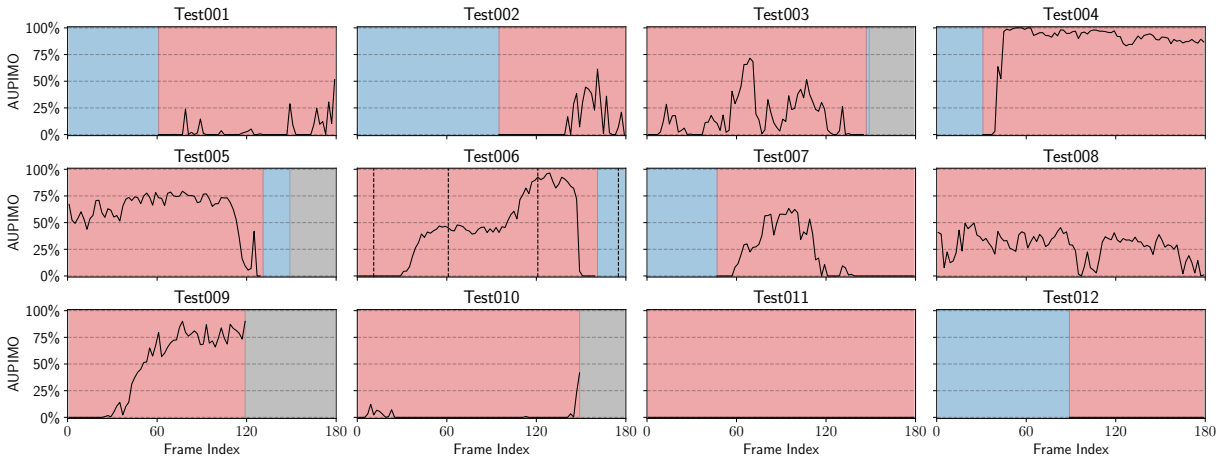


Figure 2.20: Time vs. AUPIMO in test videos from the UCSD Pedestrian dataset. The x-axis is the frame index in each video and the y-axis is the AUPIMO score at that frame. Blue zones indicate the frame is normal, red zones indicate the frame has an anomaly, and gray zones indicate there is no frame. Vertical dashed lines in "Test006" correspond to the frames shown in Fig. 2.21.

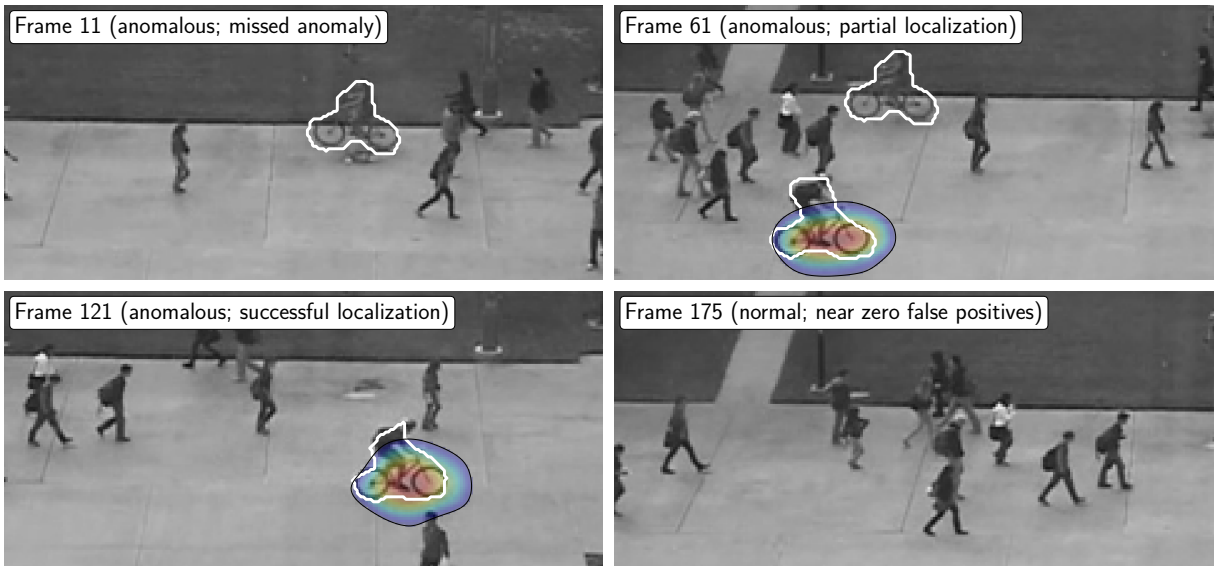


Figure 2.21: Frames from the video Test006. White contours indicate the ground truth anomalous regions. Black contours correspond to the level sets in each anomaly score map $\mathbf{a}$ at t_L, where $\mathrm{F}_{\mathrm{sh}}(t_L) = L$. Anomaly scores above below t_L are not shown and above are colored using the JET colormap with local maxima in red and t_L in blue.

6 Conclusion

We introduced area under the per-image overlap (AUPIMO): a novel recall metric tailored for ALOC 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, fine-grained performance analysis, and resilience against noisy annotation.

Evaluating eight recent models on 27 datasets with AUPIMO revealed a significant insights about the SOTA in VAD. 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?

Other Domains We focused on (2D) static image applications, but AUPIMO can be easily adapted to 3D imaging (*e.g.* X-ray tomography) and 3D point clouds (*e.g.* LiDAR). A simple extension to video applications was presented, but a more thorough adaptation could be carried out to, for instance, account for consistency and summarize the scores along the temporal dimension. Other domains like times series would require more careful adaptation, which is left for future work.

Limitations As a recall metric, the notion of segmentation quality is not covered by AUPIMO, but Chapter 3 presents an alternative to address this issue. Our proposed approach also does not cover logical anomalies as in [Bergmann, 2022] because their annotations follow a set of rules specific different types of anomalies (*e.g.* missing or extra parts).

Simplification to a Single Threshold While AUPIMO scans a range of thresholds to assess anomaly localization, a simplified single-threshold approach can be defined. This eliminates the integration in Eq. (2.13) and focuses on a specific operating point on the PIMO curve. By defining a "meaningful FPR budget", practitioners can directly determine the corresponding recall. This enhances efficiency and simplifies interpretation – aligning with practices in domains like biometrics – while maintaining all the advantages of AUPIMO and makingt it faster to compute.

Chapter 3

Unsupervised Threshold Selection for Anomaly Score Maps

Abstract

This chapter proposes a solution for inference-time threshold selection for anomaly score maps to localize anomalies in the input images. This problem is generally overlooked in the literature, and we argue that extracting statistics from set of anomalous samples breaks the unsupervised nature of the model, so we propose an unsupervised alternative. Our intuition is that the contours of local anomalies are visually identifiable in the input image, and that they should match the contours of the level sets in the anomaly score maps. We leverage the information from superpixels (an over-segmentation of the input image) to find such level sets. Our experiments on MVTec-AD show that our proposed post-processing method yields good results on different models. While the final performance is limited by the model's imperfections, we show that our method can well approximate the best achievable result on a given image.

Résumé

Ce chapitre propose une solution pour la sélection de seuils en temps d'inférence pour les cartes de score d'anomalie afin de localiser les anomalies dans les images d'entrée. Ce problème est généralement négligé dans la littérature et nous soutenons que l'extraction de statistiques à partir d'un ensemble d'échantillons anormaux rompt la nature non supervisée du modèle, donc nous proposons donc une alternative non supervisée. Notre intuition est que les contours des anomalies locales sont visuellement identifiables dans l'image d'entrée, et qu'ils devraient correspondre aux contours de niveau dans les cartes de score d'anomalie. Nous exploitons les informations des superpixels (une sur-segmentation de l'image d'entrée) pour trouver ces contours de niveau. Nos expériences sur MVTec-AD montrent que la méthode de post-traitement proposée donne de bons résultats sur différents modèles. Bien que la performance finale soit limitée par les imperfections du modèle, nous montrons que notre méthode peut bien approximer le meilleur résultat réalisable sur une image donnée.

Contents

1 Introduction

Most VAD models in the current literature infer anomaly score maps as output. These maps are one-channel images with pixel-wise scores, where higher values indicate "more anomalous" regions. Differing from common classification/segmentation models, the range of values in the anomaly score maps is commonly lower-bounded but does not have an upper bound. For instance, [Rippel, 2021c; Defard, 2021; Gula, 2023; Roth, 2022] use the notion of distance in a feature space to compute the anomaly scores, [Deng, 2022; Tien, 2023] use the notion of dissimilarity between feature vectors, and [Bergmann, 2020; Batzner, 2024] use the reconstruction error (*i.e.* a difference in the input domain).

It is common to normalize these maps and color-code them to create heatmaps, which are then used to localize the anomalies in the input images. Fig. 3.1 shows two images from the test set Screw from MVTec-AD with a min-max normalization set with different scopes: in-image and cross-image (statistics from all test samples). Notice how the visual perception of the model's behavior varies greatly depending on the choice of normalization.

As discussed in Chapter 2, the positive class (anomalous) in AD is not modeled at training time, and it is assumed that no prior information is available. Given this constraint, we argue that visualizing a model's output should not carry information from a set of known anomalous samples; thus a cross-image normalization is not suitable. Moreover, Fig. 3.1 shows that in-image normalization can also lead to misleading interpretations because the normal image seems to have a higher anomaly score than the anomalous one.

This illustrates how interpreting the heatmaps – especially when comparing different instances – is not straightforward. While different normalization strategies and other post-processing steps are commonly used, this topic is generally overlooked. This limits its practical utility to asses the model's behavior and makes it difficult to leverage the model's output in a real-world application. We believe that this leads to a more general issue of how to extract the most useful information from an anomaly score map at inference time. To the best of our knowledge, this practical problem has been given little attention in the literature, but [Tailanian, 2021; Tailanian, 2024; Jiang, 2024] are worth mentioning exceptions.

[Jiang, 2024] proposed an anomaly score threshold selection based on the model's behavior when synthetic anomalies are added to the input images. A threshold is selected based on the

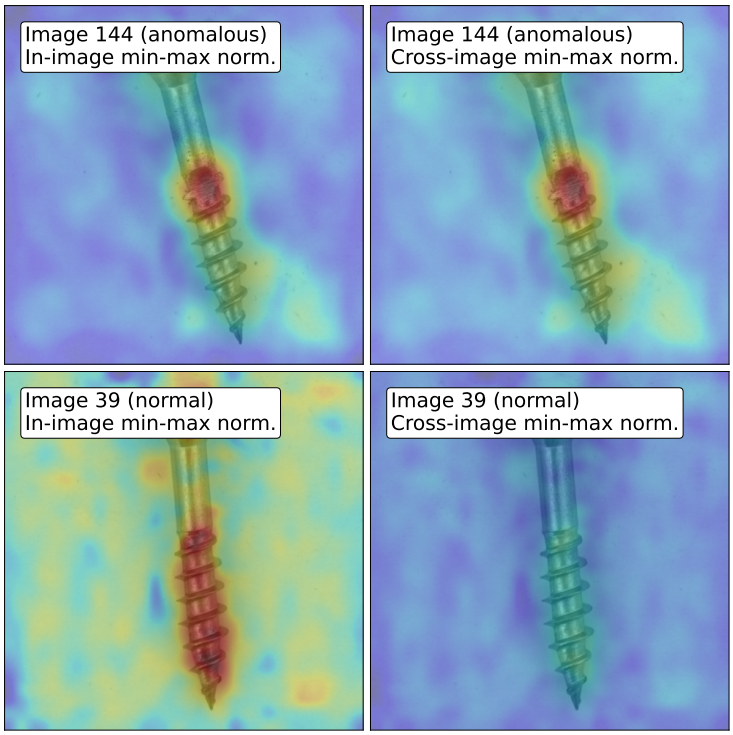


Figure 3.1: In-image vs. cross-image min-max normalization of anomaly score maps (PatchCore on MVTec-AD / Screw). Input images are superimposed with the anomaly score maps (jet colormap) with different types of normalization. Each row corresponds to a different image with the same anomaly score map colored in two different ways. Left column: in-image normalization (min and max are computed within each image). Right column: cross-image normalization (min and max are computed across the two images). In-image normalization highlights the prediction behavior within each image. While it provides a clean visualization in image 144, it produces a noisy heatmap in image 39 (normal background in yellow and green tones). Cross-image normalization highlights enables a proper comparison of the two images. Image 39 is clearly less anomalous than image 144 (which looks the opposite in the left column), but the background noise in image 144 increases. At inference time, choosing the right normalization can be challenging and bad choices can lead to misleading interpretations.

optimal IOU between the binarized anomaly score map and the ground truth segmentation of a synthetic anomaly added to an image at inference time. Their method does not rely on any cross-image statistics, and each image's threshold is selected independently. However, this process requires processing the same image twice.

[Tailanian, 2021; Tailanian, 2024] proposed strategies based on the *a contrario* framework [Desolneux, 2008]. This strategy is analogous to a statistical hypothesis test: a null hypothesis fixes a reference model to compute the probability of observing an event (*i.e.* a region in the anomaly score map being anomalous); then, if the probability is below a threshold (*i.e.* the event is "too" unlikely), the region is considered anomalous. In [Tailanian, 2024], for instance, the anomaly score map comes from a flow-based model (see Section 3.1); their choice of base distribution (in the normalizing flow) leads to the assumption that the pixels in the anomaly score map are identical and independent Chi-Squared distributed. Building upon this assumption, one can compute the probability of observing a level set (in the anomaly score map) as a function of

the anomaly score threshold and the level set's area. The advantage of using this strategy is that one can directly control the rate of false positive regions at inference time, like a "budget" of making mistakes. This notion relates to the intuition in AUPIMO's validation (see Chapter 2), although the latter is a simpler version of this idea.

This chapter proposes an alternatvie solution to inference-time threshold selection. Our goal is to extract the most useful output from a fixed model, keeping in mind that the primary utility of anomaly score maps is to localize anomalies in the input images. We introduce a model-agnostic post-processing method to transform the anomaly score maps into segmentation masks as illustrated in Fig. 3.4b. The goal is to accomplish this in an unsupervised manner by leveraging only the information contained within the image. Our method does not recourse to statistics derived from a few or synthetic anomalous images.

The chapter is organized as follows. First, we establish *image-scoped* intersection over union (iIOU) as measure of segmentation quality (Section 2), then show that inferring binarization thresholds on a per-image basis can significantly improve the visual results at inference time (Section 3). Our approach is based on the intuition that the contours of local anomalies are visually identifiable, and the contours of the level sets in the anomaly score maps should match their boundaries. We leverage the information from superpixels (an over-segmentation based on low-level image features) to find such level sets by selecting thresholds (Section 4). Finally, we adopt the validation scheme from Chapter 2 to select a minimum threshold based solely on normal samples (Section 5). This final step largely reduces the computational cost and limits the model's behavior on normal samples.

Our experiments on multiple datasets show that our proposed post-processing method yields good results on different models. While the final performance is limited by the model's imperfections, we show that our method can well approximate the best achievable result on a given image.

2 Intersection Over the Union (IOU) as Segmentation Quality Metric

The initial step in our methodology is to establish a metric to evaluate the quality of a segmentation mask. We assume that the output of a model, an anomaly score map, is fixed and we wish to extract a binary segmentation mask from it. Assuming no further post-processing, the segmentation quality only depends on the choice of the binarization threshold. Therefore, a limitation is imposed by the model's behavior: one cannot expect a perfect segmentation if such contour is not present in the anomaly score map. With this in mind, we also establish a reference segmentation quality to compare to (Section 3).

We stress that the assumptions made here intend to keep the methodology general and simplify the processing pipeline. However, our proposed pipeline can be extended to include apriori information like minimal size of the anomalies, or to include post-processing steps like morphological operations. Notice that it is common in the literature to present the anomaly score maps as heatmaps, but we argue that the visual interpretation of these maps is not

straightforward and it changes depending on the normalization scope (*cf.* Fig. 3.1).

We wish to optimize the inferred mask by selecting an optimal threshold via a guiding metric, thus the choice of the latter is crucial and must be visually meaningful. Following the discussion in Section 3.3 from Chapter 2, we focus on image-scoped metrics because they take the images' structure into account, are fast to compute, and robust to noisy annotation.

A first naive approach would be to use the *image-scoped* precision (iPrecision), which can be defined as (refer to the notation in Section 2 from Chapter 2):

$$P_{\text{i}} : t \mapsto \frac{|(\mathbf{a} \geq t) \wedge \mathbf{y}|}{|(\mathbf{a} \geq t)|} \qquad \forall t \in \mathbb{R} \text{ such that } |(\mathbf{a} \geq t)| > 0 \quad , \tag{3.1}$$

and undefined otherwise. Fig. 3.2a shows examples of the function defined above from two anomaly score maps from the Metal Nut dataset from MVTec-AD. Fig. 3.2c shows the corresponding input images with their ground truth segmentations (black dashed contour) and anomaly score maps (jet colormap normalized to the range of scores of the image). Two reference points are marked on the precision curves in Fig. 3.2a, and their corresponding segmentations are shown as solid white lines in Fig. 3.2c. The contours "A" and "B" refer, respectively, to the reference points with lower and higher threshold values (x-axis).

It can be seen that optimizing for precision is not a viable option. The precision curves are not smooth, and optimizing this metric does not correspond to improving the visual aspect of the segmentation. As seen in the examples, they show a recall-collapse phenomenon along a precision-plateau region. Increasing the threshold generally increases the precision but, at some point, it causes a strong drop of recall.

In image 11 the breakpoint is between 60% and 62% precision (*cf.* the reference points in Fig. 3.2a). Between these two points the segmentation switches from being too big to too small (recall drops from 84% to 6%). Their respective contour lines show that, visually, the results dramatically change while the precision only increases by 2%. In image 67 the same phenomenon happens between 95% and 98% precision. Recall drops from 75% to 8% respectively, and the contour lines show a similar catastrophic change.

Notice that the recall-collapse happens at different precision ranges in the two images, so the P_{i} signal alone is not enough to understand when it happens. We observed that this behavior is common across all models and datasets we tested, therefore we conclude that P_{i} is not suitable for the threshold optimization.

The iIOU, also known as Jaccard index, is defined as

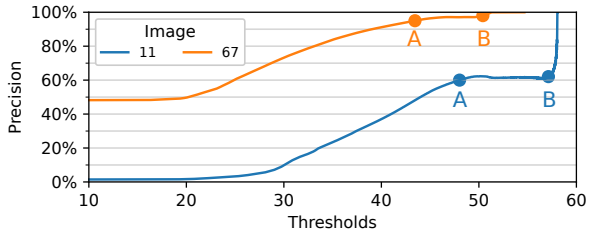
$$\text{IoU}_{\text{i}} : t \mapsto \frac{|(\mathbf{a} \geq t) \wedge \mathbf{y}|}{|(\mathbf{a} \geq t) \vee \mathbf{y}|} \quad , \tag{3.2}$$

and shows a more stable behavior. The IOU balances a trade-off between precision and recall. The F_1 score (their harmonic mean) could also be used here, but we prefer to use the former because it is easier to interpret.

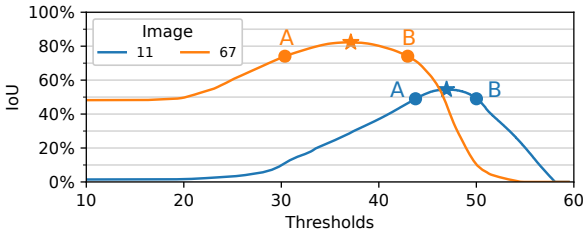
Figs. 3.2b and 3.2d show, respectively, the IOU curves and visualizations at three reference points in the same images as before. The reference points are choosen at the maximum IOU

value in the curve, and at two other points where the IOU is near-optimal with lower and higher thresholds. The IOU curves are generally smooth and show a global maximum occuring at a local maximum – multiple local maxima rarely occur. Notice how the near-optimal contours in Fig. 3.2d are visually similar to the optimal one.

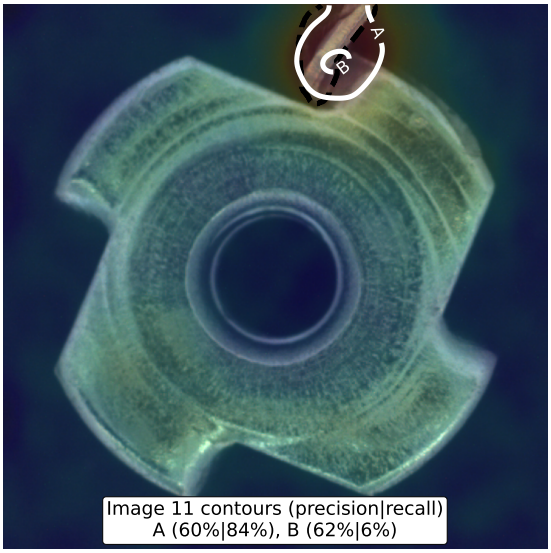
For the rest of this chapter, we use the iIOU as a direct proxy for the segmentation quality.

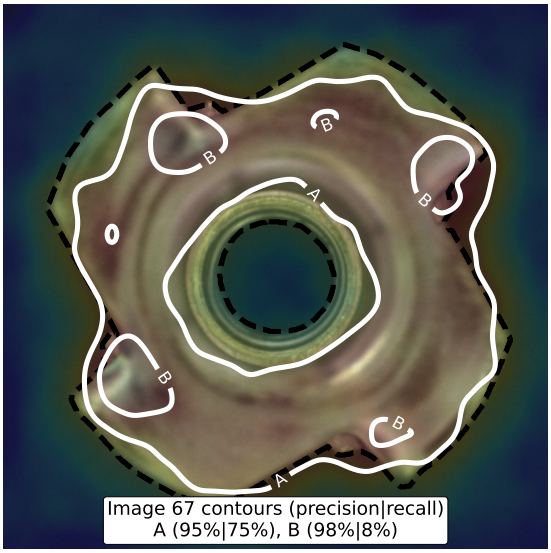


(a) Threshold vs. precision curves.

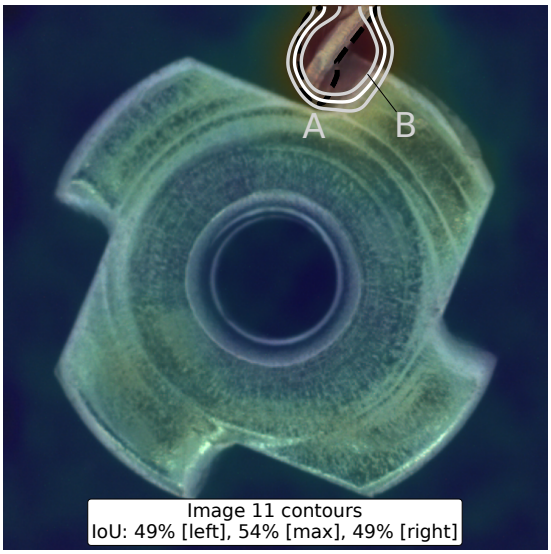


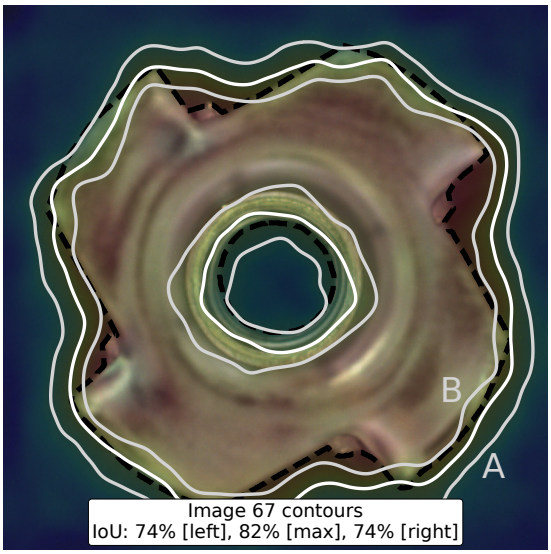
(b) Threshold vs. IoU curves.





(c) Visualization of contours at reference points on the **precision** curves.





(d) Visualization of contours at reference points on the **IoU** curves.

Figure 3.2: Precision vs IoU curves (PatchCore on MVTec-AD / Metal Nut). (a) Precision and (b) IoU are shown as functions of the binarization threshold in two anomaly score maps; reference points are marked for visual comparison. In (c) and (d), the input images are superimposed with their ground truth segmentations (dashed black line) and their anomaly score maps (jet colormap in the range of scores of the image). (c) Contours of the segmentations at the reference points A and B are shown as white solid lines. (d) Contour at the optimal IOU (star in b) is in white and the near-optimal ones (A and B in b) are in gray.

3 Global vs. Per-image Thresholds

This section has two goals. First, we define a reference segmentation performance to compare to, which consists of the best achievable result on a given image. Second, we show that an inference strategy based on a single global threshold is suboptimal. Instead, we advocate for a per-image threshold strategy and discuss its implications.

Given the $\mathrm{IoU_i}$ curve from a single image (subscript "i"), we define the *oracle threshold* t_i^* as the anomaly score threshold yielding the maximum iIOU (*i.e.* the optimal segmentation for that image):

$$t_\mathrm{i}^* = \operatorname*{argmax}_t \ \mathrm{IoU_i}(t) \quad . \tag{3.3}$$

Given all the $\mathrm{IoU_i}$ curves from a set (subscript "s") of ground truth annotations of anomalous images $\mathcal{Y}^1$, we define the average iIOU as

$$\overline{\mathrm{IoU_s}} : t \mapsto \frac{1}{|\mathcal{Y}^1|} \sum_{\mathbf{y} \in \mathcal{Y}^1} \mathrm{IoU_i^{\mathbf{y}}}(t) \quad , \tag{3.4}$$

where $\mathrm{IoU_i^{\mathbf{y}}}$ refers to the $\mathrm{IoU_i}$ curve from the specific instance $\mathbf{y}$ – its corresponding $\mathbf{a}$ is ommitted in the notation for simplicity. The *global oracle threshold* t_s^* of that same set is the anomaly score threshold corresponding to the maximum point on $\overline{\mathrm{IoU_s}}$:

$$t_\mathrm{s}^* = \operatorname*{argmax}_t \ \overline{\mathrm{IoU_s}}(t) \quad . \tag{3.5}$$

The respective IOU values at these thresholds are denoted as $\mathrm{IoU_i^*} = \mathrm{IoU_i}(t_\mathrm{i}^*)$ and $\overline{\mathrm{IoU_s}}^* = \overline{\mathrm{IoU_s}}(t_\mathrm{s}^*)$ respectively. We stress that acknowledging the oracle thresholds requires the ground truth segmentation, so they are not directly usable in practice. However, they are important building blocks for our experiments because they provide references of achievable performance to compare to.

The term "global" refers to a single threshold choice applied to all images. When not specified (as in "oracle threshold"), we refer to a threshold chosen *per-image*. In practice it is common to choose a global threshold for inference. The global oracle t_s^* gives a reference value for that choice because it yields the best achievable average performance for a given set of anomalous images. Although this methodology is common, we advocate for a per-image threshold strategy instead.

Fig. 3.3 illustrates, on the dataset Transistor from MVTec-AD, how the per-image strategy can yield significant improvements compared to a global strategy. Fig. 3.3a shows, on the left, the $\overline{\mathrm{IoU_s}}$ curve for the whole test set. Cross-image percentiles – pk denotes the k-th percentile – of the iIOU values are also shown as curves for comparison. The optimal point of the average curve $\left(t_\mathrm{s}^*,\ \overline{\mathrm{IoU_s}}^*\right)$ is marked with a circle on the curve. The percentiles show that the choosing a global threshold leads to a large variance in the segmentation quality. At the average-optimal point, for instance, p5 and p95 are, respectively, 15% and 75%.

Fig. 3.3a shows, on the right, four (color-coded) $\mathrm{IoU_i}$ curves with their optimal points marked as circles on the curves. The global oracle threshold t_s^* is shown for reference. Enabling a per-

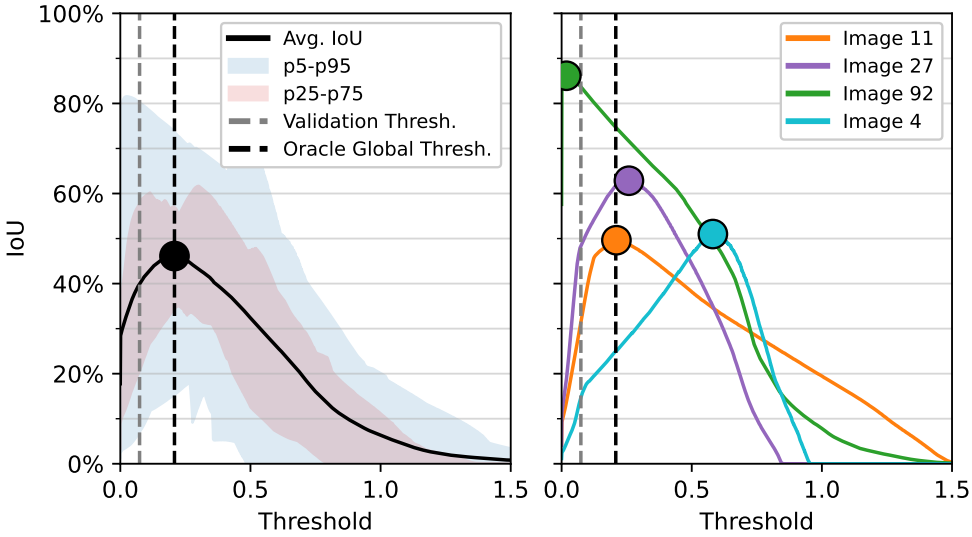
image threshold inference can lead to significant visual improvements at inference time, thus making better use of the model's potential.

Fig. 3.3b shows how this change of strategy corresponds indeed to visually meaningful improvements in the segmentation quality. The input images are superimposed with the ground truth segmentation (dashed black line), t_{s}^{*} (white solid line), and t_{i}^{*} (solid lines colored according to the colors of their respective curves in (Fig. 3.3a)). Images 11, 27, 92, and 4 show improvements progressively more significant.

We quantify these improvements across multiple datasets and models in our experiments (Section 6). Fig. 3.8 confirms that, on average, using a single global threshold for inference leads to strongly suboptimal results.

Oracle Threshold as Evaluation Metric $\mathrm{IoU}_{\mathrm{i}}^{*}$ is the best approximation of the ground truth segmentation that can be extracted from a given anomaly score map. The model's (imperfect) behavior limits the optimality of the oracle threshold, thus $\mathrm{IoU}_{\mathrm{i}}^{*}$ can also be used to evaluate the underlying model itself. The statistics (aggregated across images) in Section 6 can be used to assess cross-model and cross-dataset performance as a complement to the benchmark presented in Chapter 2.

While this change of paradigm (per-image instead of single threshold) is simple, it has been overlooked in the literature. Common evaluation metrics in VAD implicitly assume a single threshold for all images. That is the case, for instance, for the instance average precision (IAP) [Zhang, 2023c] and for the F_1-max [Zou, 2022; Jeong, 2023]; they are discussed in Section 2 from Chapter 2. Our observation suggests that such metrics largely underestimate the potential of the models to produce visually meaningful segmentations.



(a) Oracle thresholds on the threshold vs. IOU curves.

(b) Visualizations of the level sets' contours at the oracle thresholds.

Figure 3.3: Oracle thresholds: global vs. per-image (EfficientAD on MVTec-AD / Transistor). (a) The average IOU curve (black) for the whole test set is shown on the left, and the IoU_i curves (colorful) for four images are shown on the right. The optimal points of the average curve and the per-image curves are marked with circles. The validation threshold is shown in a vertical gray line. (b) The input images are superimposed with the ground truth segmentation (dashed black line) and the masks yielded by the thresholds seen in a: the global oracle (white solid line), the per-image oracle (solid lines colored as their respective curves in a), and the validation threshold (gray line).

4 Superpixel-based Heuristic Threshold Selection

We propose an unsupervised threshold selection based on the intuition that the contours of local anomalies are visually identifiable. Therefore, a low-level description of the image should be able to capture these contours. Based on this idea, our method consists of finding level sets in the anomaly score map that match the boundaries of superpixels.

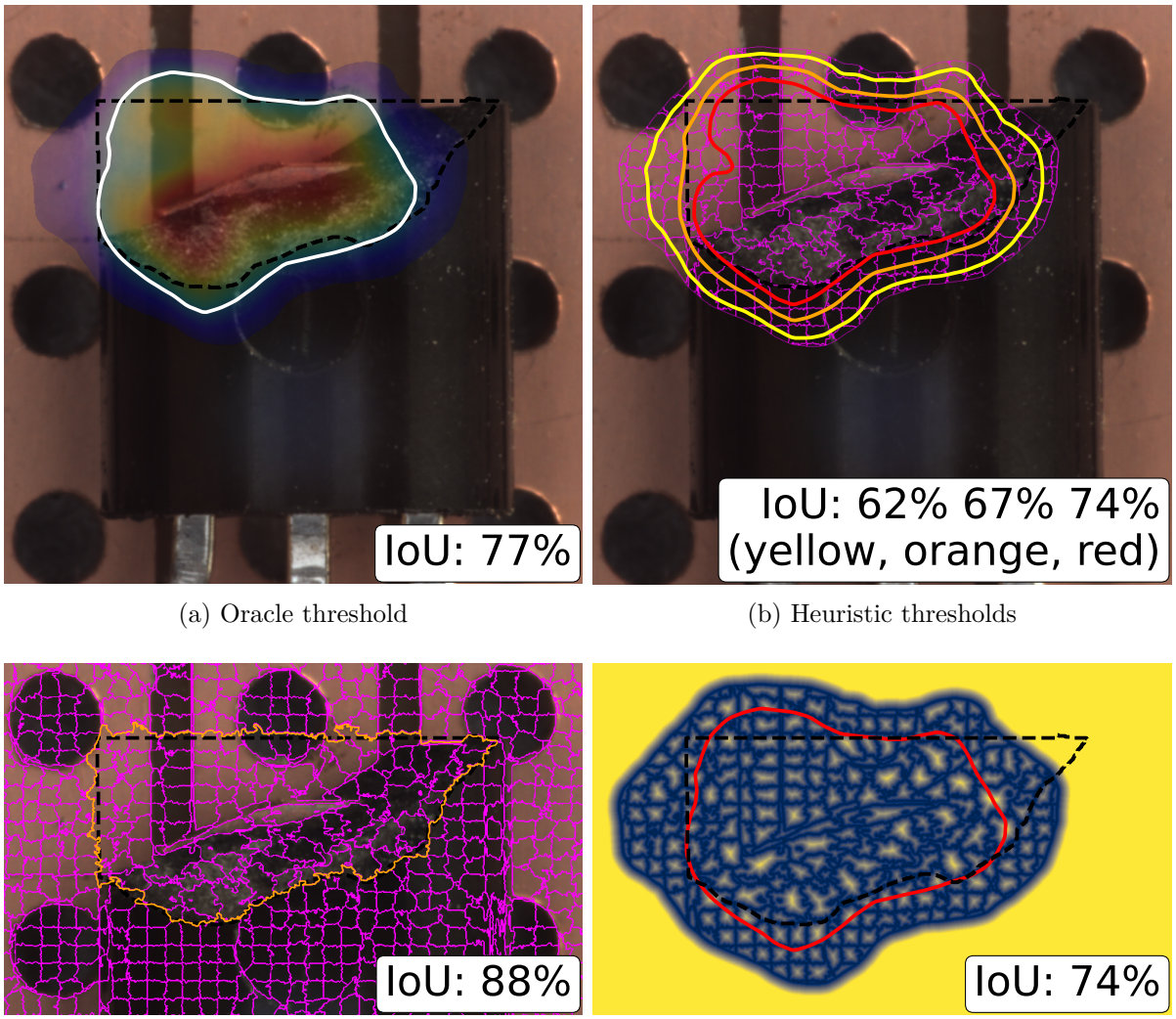
Superpixels Our approach is to use superpixels as a low-level representation of the image. Superpixels – labels obtained from an oversegmention – provide a compact and low-level representation of the image. From this label image, we extract a binary boundary mask $b \in \{0,1\}^M$, where M is the image resolution (simplified into a single dimension) and "1" means "is a boundary pixel". A boundary pixel is defined as a pixel that has at least one neighbor (connectivity of 1) with a different label. An example is shown in Fig. 3.4c, where magenta lines mean "1" in b (the superpixels' boundaries).

Oversegmention Algorithms Multiple algorithms can be used to obtain superpixels. We experimented with Felsenszwalb's efficient graph-based segmentation [Vedaldi, 2008], simple linear iterative clustering (SLIC) [Achanta, 2012], quick shift [Felzenszwalb, 2004], and compact watershed [Neubert, 2014]. We found that compact watershed [Neubert, 2014] provides good results across multiple datasets and models with a reasonable computational cost.

Compact Watershed The marker-controlled watershed [Beucher, 2018] treats the input image as a topographic surface, where the intensity values represent elevations. The spatial gradient – in our case, a Sobel filter [Kroon, 2009] is used – is applied to the grayscale input image. The watershed segmentation works by flooding the gradient image from its minima and separating regions using the boundaries where flooding waters from different minima meet. In the compact watershed [Neubert, 2014], a uniform grid is used to start the flooding process and a term, called *compactness*, is added to the elevations. The compactness term measures the distance to initial point of flooding, so higher values lead to more regular-shaped superpixels. We use it with compactness set to 10^{-4} and 3333 grid-spaced markers (corresponding to superpixels of relative size of 3×10^{-4}).

Do Superpixels Capture the Anomalies' Contours? To validate the intuition that the anomalies' contours are captured by the superpixels, we define an oracle superpixel selection. We compute the best achievable iIOU by selecting an optimal subset of superpixels. A best-first search algorithm is used to select the superpixels that maximize the IOU. The search is initialized with the superpixels entirely contained in the ground truth segmentation. At each step of the search, the superpixel yielding the largest increase in IOU is added to the selection. The search stops when the IOU stops increasing or when there are no more superpixels touching (*i.e.* overlapping with) the ground truth segmentation. Fig. 3.4c shows an example of oracle superpixel selection (orange contour). It achieves an IOU of 88%, which is actually higher than that from the oracle threshold (77%, in Fig. 3.4a). In Section 6 we show quantitative results

across multiple datasets and models. Fig. 3.8 shows that, on average, compact watershed does a good job at capturing the anomalies' contours. The more detailed analysis from Fig. 3.7 reveals, however, that this approach particularly struggles with some datasets. Finally, it is also worth noting that the superpixels also capture the boundaries of other, normal structures in the image, thus they are not enough – alone – to identify the anomalies.



(a) Oracle threshold

(b) Heuristic thresholds

(c) Oracle superpixel selection

(d) Superpixel-boundary distance map

Figure 3.4: Oracle vs. heuristic thresholds (Efficient-AD on MVTecAD / Transistor). Ground truth segmentation (dashed black line) is superimposed on all images for the sake of comparison. (a) Oracle threshold (white solid line). (b) Heuristic thresholds (yellow, orange, and red solid lines) and valid superpixels' boundaries (magenta solid lines). (c) Oracle superpixel selection. Contours in magenta are the boundaries of the superpixels, and the orange line is the contour of the oracle superpixel selection. (c) Superpixel-boundary distance map (cividis colormap; blue is 0 and yellow is 1) and best heuristic segmentation (red solid line).

Level Sets vs. Superpixels Given the aforementioned boundary mask b, our initial intuition is translated into finding a level set (in the anomaly score map) that matches the boundaries of the anomalies' superpixels. The target superpixels (*i.e.* those belonging to the anomalies) are not known in advance, and no prior information about the anomaly score map is used. The core idea of our heuristic is that both sources of information ($\mathbf{a}$ and b) share a common structure containing the anomalies' contours. Therefore, our goal is to find a threshold such that its level set' shape is similar to a subset of the superpixels. We use the notion of contour distance to measure the similarity between level sets and superpixels. Fig. 3.4b shows examples of level sets that locally minimize the contour distance to the superpixels' boundaries. The oracle IOU (Fig. 3.4a) is reasonably approximated and their visual aspect are close. The next paragraphs detail how these level sets are selected.

Boundary Distance Map The Euclidean distance transform (EDT) is applied to b to obtain a boundary distance map $d \in \mathbb{R}_+^M$, where each pixel value is the distance to the nearest boundary pixel. The latter is then normalized by the maximum distance in d, and the normalized distance map is noted $\hat{d} \in [0,1]^M$ An example of $\hat{d}$ is shown in Fig. 3.4d, although it is modified with an additional step introduced in Section 5.

Contour Distance Curve (Heuristic Signal) Given a threshold t and its level set, $\hat{d}$ is used to assign a distance value to all the points along its 2D path. The mean of these values is assigned as the heuristic signal associated with the threshold t. In practice, a level set is computed by thresholding the anomaly score map giving a binary mask noted $\mathbf{a}_t = \mathbf{a} \geq t$. Then, its outer contour is computed by subtracting itself from its morphological dilation with a 3×3 square, giving another binary mask noted $\mathbf{a}_t^{\mathrm{c}}$ (the superscript "c" stands for "contour"). Finally, the mean contour distance associated to t is computed as

$$h : t \mapsto \frac{\left|\hat{d} \circ \mathbf{a}_t^{\mathrm{c}}\right|}{|\mathbf{a}_t^{\mathrm{c}}|} \in [0,1] \quad , \tag{3.6}$$

where $\circ$ denotes the Hadamard product and $|\cdot|$ is overloaded to denote the sum of the values in a real-valued array[1]. Two examples of h are shown in Fig. 3.5. We build discrete curves of this heuristic signal from a sequence of 500 thresholds linearly spaced between the minimum and maximum anomaly score values of a given $\mathbf{a}$. For the sake of comparison, the heuristic curves from the images in Fig. 3.5 are shown with their corresponding $\mathrm{IoU_i}$ curves. The selected thresholds (next paragraph) and the iIOU achieved at these thresholds are shown as dots on the curves.

Local Minima The final step consists of finding the threshold values at the local minima of the heuristic signal curve. Since h is discrete, in practice we use the relative minima function[2] with order – number of points (at each side) along the curve used for comparison – set to ten,

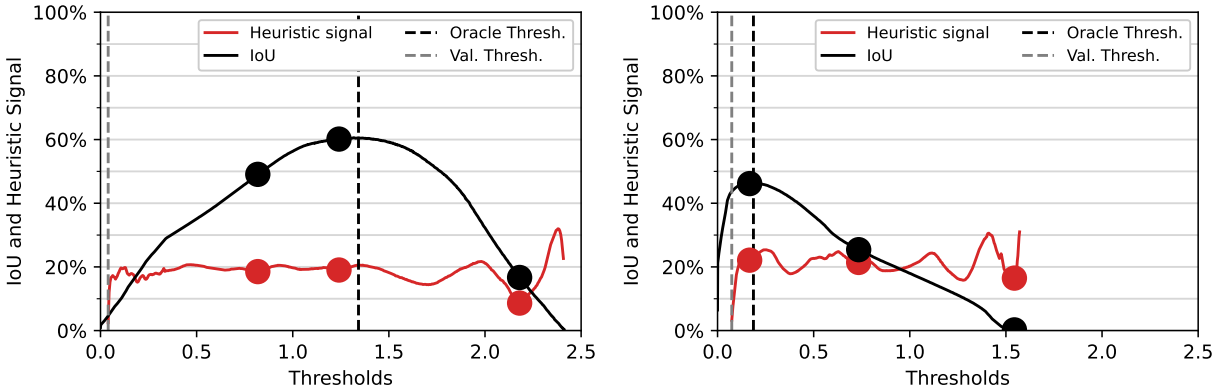
[1] Note that, when the array is binary, this definition simplifies to the one previously introduced in Table 2.1 from Chapter 2

which represents 2% of the curve length.

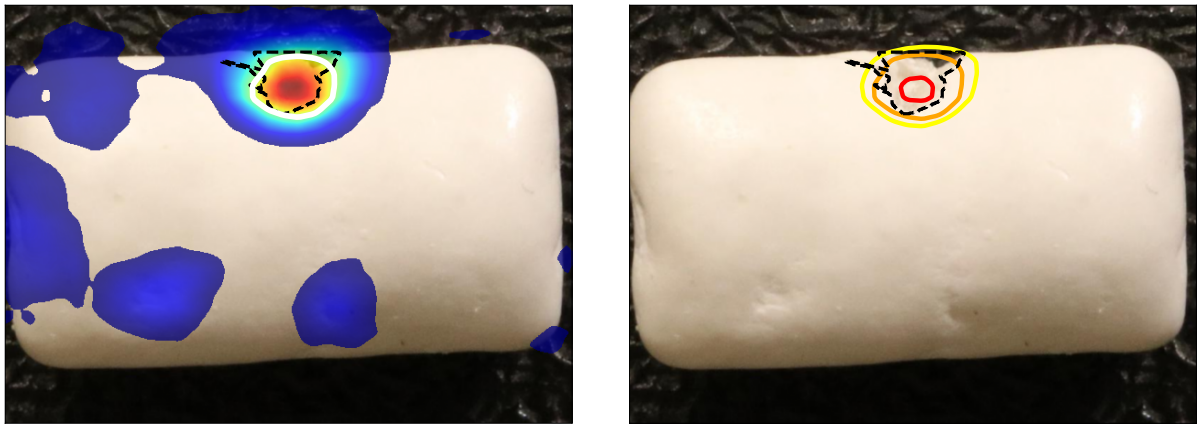
Inference As a post-processing method, our threshold is meant to proactively propose segmentation contours to the user as a complement to the model's output (*i.e.* **a**). While this heuristic generally finds meaningful contours, it suffers from a lack of robustness like the smallest contour in Fig. 3.5c. With this in mind, we propose that several heuristic thresholds are selected (*i.e.* several local minima), leaving the choice of the best segmentation to the user. This strategy has a good chance of proposing at least one meaningful segmentation as shown in our experiments (Section 6). The final output consists, by default, of up to the three thresholds (and their corresponding contours) with the lowest heuristic signal values. When less than three local minima are found, all of them are proposed.

Resolution To mitigate discretization issues, the superpixels are computed from the input image with doubled resolution – each dimension is doubled, thus the number of pixels is quadrupled. As a consequence, all the aforementioned maps are also at doubled resolution. The anomaly score map is equally upsampled to match this resolution. Both the image and the anomaly score map are upsampled with bilinear interpolation.

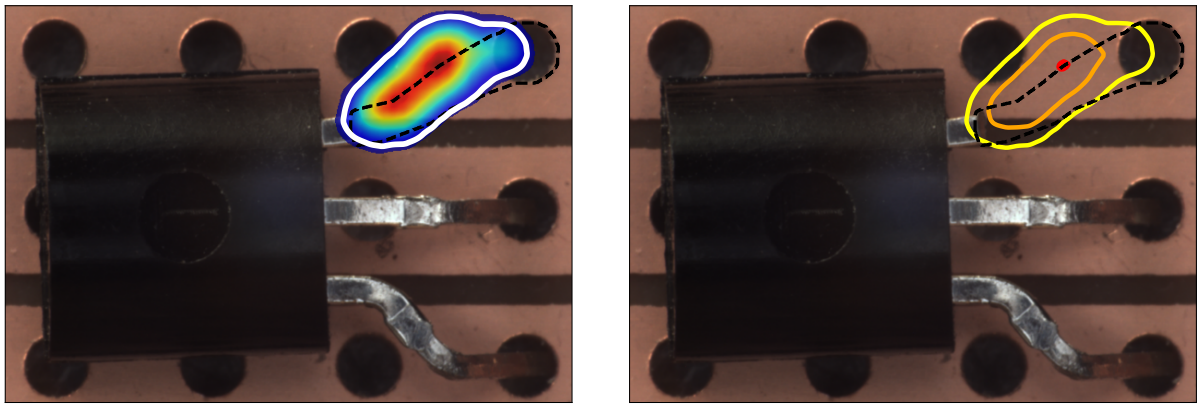
[2]Refer to `scipy.signal.argrelmin` in `scipy` [Virtanen, 2020].



(a) Heuristic vs. IoU curves. Heuristic signal in red and IoU curves in black. Red dots correspond to local minima of the heuristic signal, and black dots are their corresponding IoU values. Left: sample image from VisA / Chewing Gum, visualized in (b. Right: sample image from MVTec-AD / Transistor, visualized in (c).



(b) VisA / Chewinggum



(c) MVTecAD / Transistor

Figure 3.5: Heuristic threshold selection. (b) and (c) left: input image superimposed with its valid anomaly score map, ground truth segmentation (dashed black line) and the best heuristic segmentation (solid white line). (b) and (c) right: input image superimposed with the ground truth segmentation (dashed black line) and three heuristic threshold choices (solid colorful lines).

5 Validation Threshold

We select a minimum threshold with a validation strategy as in Chapter 2 – further referred to as *validation threshold* t_v. The latter is chosen based on a maximum tolerance to false positives on normal images, thus not requiring positive (anomalous) samples. As discussed in Section 3.1 in Chapter 2, this is important to not bias our method towards a limited set of anomaly types. The criterion used here is that t_v is such that the average iFPR on a set of unseen normal images is of 1%. Since the evaluation in this chapter only concerns anomalous images, we borrow the normal images from the test sets of the datasets used in our experiments.

We further refer to "valid regions" as those where $\mathbf{a} \geq t_v$, and "valid anomaly score maps" as a modified version of $\mathbf{a}$ such that values outside the valid regions are set to undefined. We similarly refer to "valid superpixels" as those that overlap with the valid regions.

The parameter t_v is a global threshold as defined in Section 3, but it used as a lower bound for the actual thresholds yielded by the heuristic. Fig. 3.3 shows examples of a validation threshold on the IOU curves and how it compares to the oracle segmentations. This approach provides a first estimate of interesting regions and alleviates the computational cost of the heuristic. The segmentation contours at t_v is not important because a higher threshold is chosen later.

When using this validation strategy, the normalization of the boundary distance map only accounts for the valid regions, and $\hat{d}$ is clipped to the range $[0, 1]$. At inference time, any threshold chosen at the post-processing stage is lower-clipped by the validation threshold. This clipping eventually leads to sub-optimal segmentations. Image 92 in Fig. 3.3 (green) shows an example where that happens – notice how $t_i^* < t_v$ For the sake of making a fair comparison, we also integrate the validation threshold clipping to the oracle IOU. Fig. 3.8 shows that the drop of performance is rather small on average.

By design, the valid zones are expected to be mostly "small" in normal images. As such, this extra step can also be used to dismiss predictions with small relative size (compared to the chosen FPR parameter).

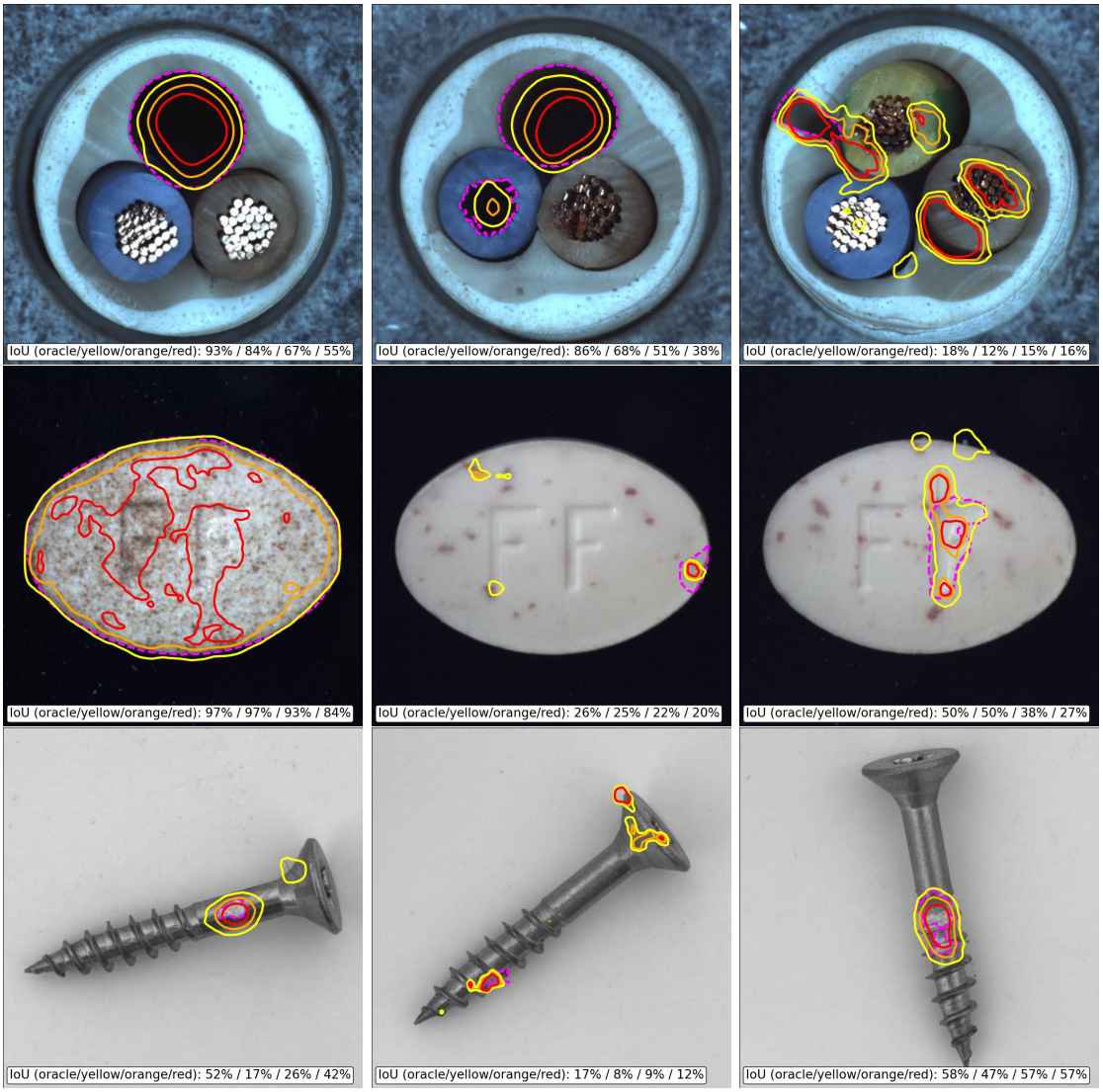


Figure 3.6: Examples of good and bad segmentations from the heuristic threshold selection. The input images are superimposed with the ground truth segmentation (dashed magenta line) and the three heuristic segmentations (solid yellow/orange/red lines). The images are from three datasets from MVTec-AD and the model is EfficientAD.

6 Experiments

Since our post-processing is totally model-agnostic, we test it on several models. We select the (overall) best four models from the benchmark presented in Chapter 2 (default versions, *cf*. Table 2.3): PatchCore [Roth, 2022], EfficientAD [Batzner, 2024], RevDist++ [Tien, 2023], and UFlow [Tailanian, 2024]. All training and inference hyperparameters are kept as in Chapter 2. All datasets from MVTec-AD and VisA were benchmarked. Recall that the input images are downsampled to 256×256 pixels, and the anomaly score maps are upsampled back to the image's original resolution with bilinear interpolation. All metrics are measured at the original resolution. We used `scikit-image`'s [Walt, 2014] implementation of compact watershed [Neubert, 2014].

Fig. 3.8 shows the distribution of iIOU achieved with the oracle superpixel selection ("SP Sel."), the (per-image) oracle threshold ("Oracle"), and the heuristic threshold ("Heuristic"). Both threshold selections account for the validation threshold clipping. The heuristic's performance is the best out of the three proposed thresholds at inference. The model used is EfficientAD on all datasets from MVTec-AD and VisA. Fig. 3.6 shows qualitative examples of good and bad segmentations from the heuristic threshold selection.

The oracle superpixel selection generally achieves the best scores, confirming our intuition that the anomalies' contours are captured by the superpixels. In some datasets like VisA / Macaroni 1 and 2, compact watershed struggles to capture the anomalies' contours, which limits the potential of the heuristic. Overall, the performance variance within dataset is large, suggesting that the method is not robust to all types of anomalies. A hyperparameter search could be conducted to improve the robustness of the method as an extension of this work.

In most datasets the heuristic threshold performance is close to the oracle threshold performance, showing that the method is able to leverage the information from the two sources (the input image and its anomaly score map). This also shows that the limitation of the method is rather imposed by the underlying model's imperfect behavior.

Fig. 3.8 shows aggregated results from all the models in our experiments. The cross-dataset averages of the (cross-image) average IOU scores are presented as horizontal bars. We compare the segmentation quality of all the strategies presented in this chapter: the global oracle threshold, the per-image oracle threshold with and without validation clipping, the heuristic threshold with validation clipping (best of three), and the oracle superpixel selection. For the global oracle threshold, the threshold is global but the IOU is computed per-image – *i.e.* the iIOU values along the black dashed line in Fig. 3.3a, right. Notice that oracle superpixel selection only depends on the input images, so the results are the same for all the models.

The results from different models have variable performance, but comparing the threshold choices for each model globally shows the same trends. As discussed in Section 3, Fig. 3.8 clearly demonstrates that adopting a per-image threshold strategy is beneficial relative to a global one. It also shows that clipping the thresholds with a validation threshold does not significantly impact the performance. Finally, the heuristic threshold reaches a reasonable approximation of the oracle segmentation; despite its limitations, it is always better than the global oracle

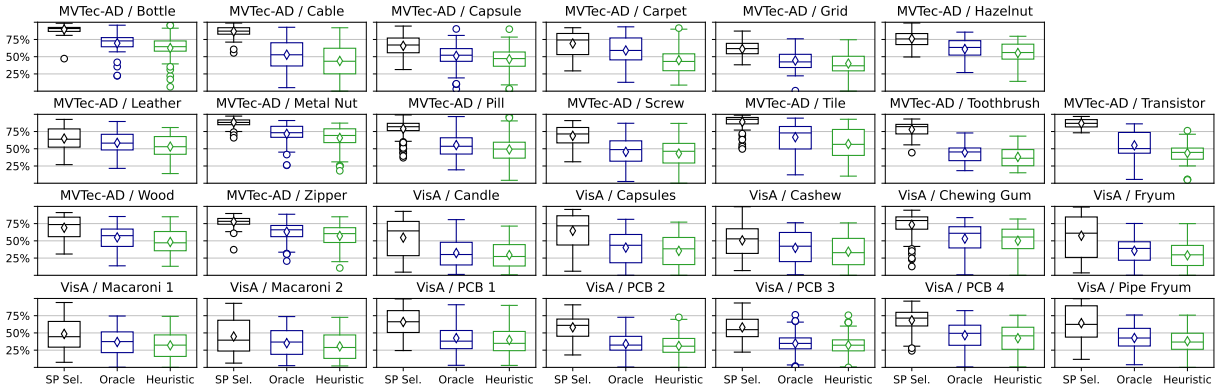


Figure 3.7: Superpixel selection vs. oracle threshold vs. heuristic threshold (best of five) (Efficient-AD on all datasets from MVTec-AD and VisA). Heuristic's reported performance is the best out of the three proposed thresholds at inference.

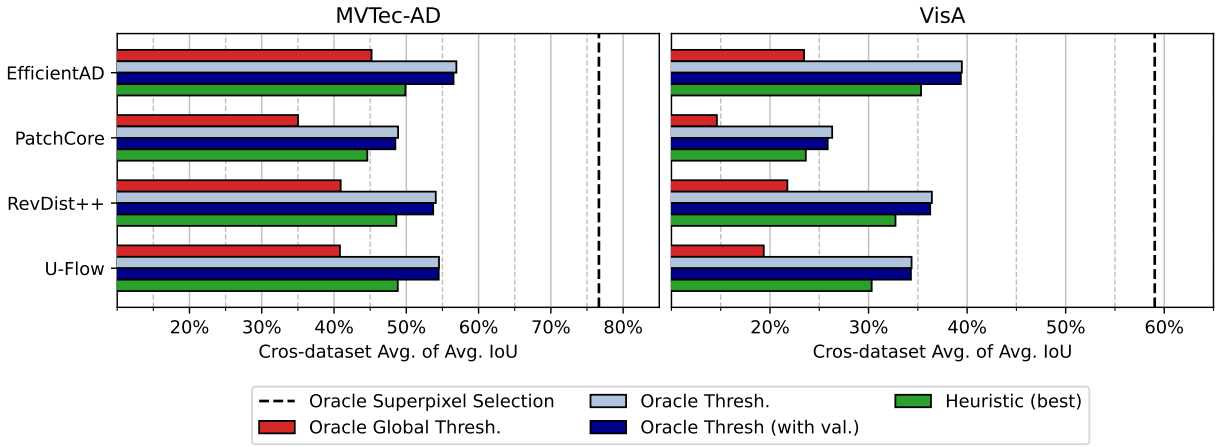


Figure 3.8: Cross-dataset average of cross-image average IOU scores for different models on MVTec-AD and VisA. Heuristic's reported performance is the best out of the three proposed thresholds at inference.

threshold on average.

7 Conclusion

We proposed a heuristic threshold selection method for level set-based segmentation of local anomalies on anomaly score maps. Instead of using a single, globally-set threshold to binarize anomaly score maps, we proposed a per-image threshold strategy. Our results demonstrate that this simple change of paradigm can provide useful improvements for practical uses of VAD models at inference time.

The methodology proposed in this chapter is unsupervised – not depending on any knowledge of the anomalies – and model agnostic. It relies on the assumption that the anomalies' contours are present in the anomaly score maps and can also be captured by a low-level segmentation of the image. A heuristic signal is built by measuring the average distance from level sets (from a score map) to the boundaries of superpixels (the low-level segmentation). Finally, the local

minima of this signal are identified as potential thresholds where superpixels' boundaries are well approximated by the level sets.

Our post-processing strategy is novel in the context of VAD as it proposes a threshold selection completely unsupervised. Adopting a per-image selection is conceptually simple and provides a significant improvement over a global selection. This strategy has been neglected in previous works, which have mostly focused statistics aggregated across images. Our approach combines the use of superpixels with anomaly detection, leveraging the information from the input image. This integration of superpixels and anomaly detection is a unique contribution of our work, which we hope will inspire future research in the field.

The results demonstrated that our method has limitations but they can be mitigated by proposing three (instead of one) thresholds at inference time, which does not fundamentally change the usability of the method. Its role is to proactively propose meaningful contours, so the final user can visually identify the best one. Future work could focus on addressing the robustness issues as well as replacing the threshold selection with other post-processing techniques (*e.g.* morphological reconstruction) to further enhance the segmentation results. Overall, our study contributes to solving a practical challenge and improving usability of anomaly localization models.

Chapter 4

Analysis, Optimization, and Visualization of Gaussian Models

Abstract

This chapter delves into the analysis and optimization of multivariate Gaussian (MVG)-based models Section 1 introduces the notation used in this chapter and the motivation behind the study. Section 2 introduces a novel optimization strategy for MVG models, Greedy Eigencomponent Selection (GreedyES). A series of experiments evaluate its effectiveness and generalization capacity. These techniques are aimed at providing a qualitative analysis of the model's predictions and understanding the behavior of the underlying neural network. Our findings reveal the pretrained model exhibit redundant and some spurious features that can be detrimental to the performance. Contradicting existing theories, we show evidence that feature variance does not correlate with efficacy in anomaly detection. Results suggest that deeper layers can encode a class in smaller [anomaly-distinguishable] embedding than shallower layers, also contradicting previous works. Section 3 presents visualization strategies for MVG-based models.

Résumé

Ce chapitre traite de l'analyse et de l'optimisation des modèles basés sur une loi normale multivariées. Section 1 présente la notation utilisée dans ce chapitre et la motivation de l'étude. Section 2 présente une nouvelle stratégie d'optimisation pour ces modèles, Greedy Eigencomponent Selection (GreedyES). Une série d'expériences évalue son efficacité et sa capacité de généralisation. Ces techniques visent à fournir une analyse qualitative des prédictions du modèle et à comprendre le comportement du réseau de neuronnes sous-jacent. Nos résultats révèlent que le modèle pré-entraîné présente des *features* redondantes et certaines *features* parasites qui dégradent la performance. Contredisant les théories existantes, nous démontrons que la variance des *features* n'est pas en corrélée avec son efficacité en détection d'anomalies. Les résultats suggèrent que les couches plus profondes peuvent encoder une classe dans un *embedding* [adapté à la détection d'anomalie] plus petit que les couches moins profondes, ce qui va également à l'encontre des travaux précédents. Section 3 présente des stratégies de visualisation pour les modèles basés sur une loi normale multivariées.

Contents

1 Introduction

Among the various approaches to AD (see Chapter 1), probabilistic models have been widely used in various domains, including computer vision. The principle behind these models is that anomalies are unlikely to occur, thus they can be detected by measuring the likelihood of a data point under a probabilistic model. To achieve this, the learning process consists of estimating this model from a set of normal data points; at inference time, the likelihood of a new data point is computed, and a decision is made based on a threshold.

The multivariate Gaussian (MVG) distribution is a common choice of probabilistic model because it is simple and has a closed-form solution for the likelihood computation. Other options like Gaussian mixture model (GMM) (multiple Gaussian modes) and kernel density estimation (KDE) (a non-parametric model) have also been used, but we focus on the Gaussian model for

its simplicity and extensive use in the literature.

Several works have explored the use of Gaussian models for VAD [Defard, 2021; Kim, 2021; Jang, 2023; Zheng, 2022; Rippel, 2021c; Rippel, 2021a; Lin, 2022; Kim, 2021] and out-of-distribution (OOD) [Du, 2022; Lee, 2018; Kamoi, 2020]. In general, these models leverage the feature space learned by a pre-trained convolutional neural network (CNN) to represent the input images. Then, an MVG-based model is constructed from a collection of vectors.

We primarily focus on the image-wise models, where each feature vector encapsulates the characteristics of an entire image. In contrast, patch-wise models (*e.g.* PaDiM [Defard, 2021]) consider smaller parts of the image (*i.e.* patches). They are shortly explored in this chapter in the context of visualization. Both model families operate under the same principle, but the feature vectors are extracted differently, so their parameters are not directly comparable. However, we believe that future work could benefit from the insights presented here to improve patch-wise models as well.

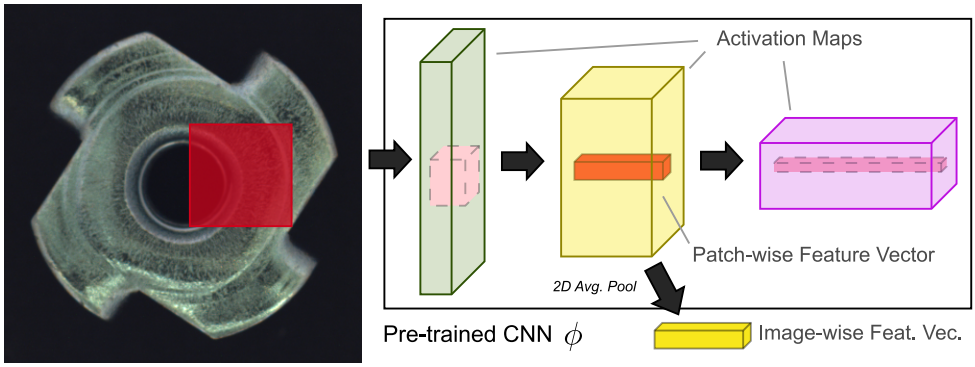
Our major motivation for this work is the simplicity of the MVG model. The low computational cost of inference and no need for retraining make it an attractive choice for plug-and-play integration with any pre-trained CNN. In other words, the results of our work could be extended to be integrated in other CNN-based applications. By adding low computational overhead and not requiring modification of the underlying CNN, an MVG model can be used in parallel to enhance the detection capabilities of a generic CNN.

Outline Section 1 begins by defining the necessary concepts and notation – *i.e.* feature extraction from a CNN, feature vectors, model parameters, anomaly score, *etc.* In Section 2, our contributions extend to previous results in the field, specially the unexpected efficacy of employing *negated* principal component analysis (NPCA) (defined in Section 1.2) for anomaly detection tasks. Leveraging the insights from our framework, we propose a training pipeline with synthetic data to enhance the anomaly detection performance of the MVG model. Finally, in Section 3, we introduce visualization techniques to interpret the model's decision.

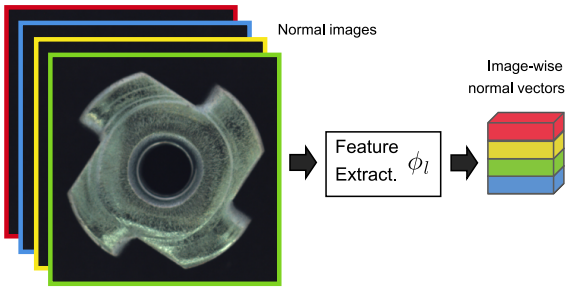
1.1 Multivariate Gaussian Distribution

The MVG models analysed in the this chapter are based on feature vectors extracted from pre-trained CNNs. We present *image-wise* and *patch-wise* models, meaning that each feature vector represents, respectively, an entire image or a patch of an image. The focus is on the image-wise case (unless stated otherwise, we refer to this model), but the patch-wise model is also discussed in the context of visualization in Section 3.1. In both cases, the CNN's parameters are frozen, so we omit them from the notation. Here we present some general concepts and definitions that are used in the following sections.

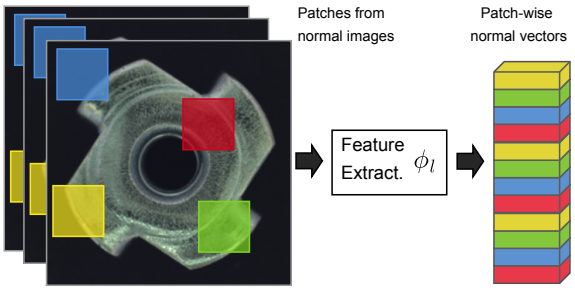
An input image $\mathbf{I}$ is passed to a backbone neural network ϕ, and the activation maps from its layer l are extracted as illustrated in Fig. 4.1a. This 3D tensor – two (discrete) spatial axes and one channel/feature axis – is further referred to as a *feature map* since these activations are interpreted as a locality-preserving representation of the input image. A 2D average pooling is applied to aggregate the spatial dimensions, outputting a *feature vector* $\mathbf{x} = \phi_l(\mathbf{I}) \in \mathbb{R}^d$, where



(a) Feature map extraction from a pre-trained CNN. Blocks of layers are represented by arrows. Feature maps (activations from the layer that produced it) are represented by cuboids.



(b) Image-wise feature vectors.



(c) Patch-wise feature vectors.

Figure 4.1: Feature extraction from a pre-trained CNN.

ϕ_l is referred to as "feature extractor" and d is the number of features/channels, *i.e.* the size of the embeddings.

An MVG model assumes that the feature vectors extracted from normal images follow a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with mean $\boldsymbol{\mu} \in \mathbb{R}^d$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$. Given this assumption, the likelihood of a data point can be computed with the probability density function of the Gaussian distribution:

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^d \det(\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\right) \quad , \tag{4.1}$$

where $\det(\boldsymbol{\Sigma})$ is the determinant of $\boldsymbol{\Sigma}$. In AD, this likelihood is then used as an anomaly score, such that data points with high likelihood are considered normal and data points with low likelihood are considered anomalous. However, it is common practice to not use the likelihood directly, but rather the notion of distance from the mean $\boldsymbol{\mu}$. Correspondingly, data points close to the mean are considered normal, and data points far from the mean are considered anomalous, as illustrated in Fig. 4.2a.

(a) Images in a feature space.

(b) Equidistant lines with the Mahalanobis distance.

Figure 4.2: Intuition from the Gaussian model and visualization of the Mahalanobis distance.

The expression in Eq. (4.1) can be rewritten as

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^d \det(\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}\,\mathrm{MahaDist}_l\,(\mathbf{x})^2\right) \quad , \quad \text{where} \tag{4.2}$$

$$\mathrm{MahaDist}_l\,(\mathbf{x}) = \sqrt{(\mathbf{x}-\boldsymbol{\mu})^T\,\boldsymbol{\Sigma}^{-1}\,(\mathbf{x}-\boldsymbol{\mu})} \tag{4.3}$$

is the Mahalanobis distance [Mahalanobis, 1936]. Notice that replacing the precision matrix $\boldsymbol{\Sigma}^{-1}$ (the inverse of the covariance matrix) in Eq. (4.3) with the identity matrix (unit variance in all axes and no correlation between them) would yield the Euclidean distance between $\mathbf{x}$ and $\boldsymbol{\mu}$. The precision matrix $\boldsymbol{\Sigma}^{-1}$ distorts the Euclidean distance to account for the correlation between the features, creating elipsoid-shaped equidistant lines from the mean (illustrated in Fig. 4.2b). This distance is used as an anomaly score, where a higher value (*i.e.* more distant from the center) indicates a higher chance of an input image being anomalous.

The mean and the covariance matrix are fitted from the a set of feature vectors $\mathcal{X}_{\text{train}} = \{\, \phi_l(\mathbf{I}) \mid \mathbf{I} \in \mathcal{I}_{\text{train}} \,\}$ extracted from the training set $\mathcal{I}_{\text{train}}$ with normal images only. The empirical mean $\hat{\boldsymbol{\mu}} \in \mathbb{R}^d$ and the empirical covariance matrix $\hat{\boldsymbol{\Sigma}}_{\text{s}} \in \mathbb{R}^{d\times d}$ are, respectively, computed from their maximum likelihood estimator (MLE)s:

$$\hat{\boldsymbol{\mu}} = \frac{1}{|\mathcal{X}_{\text{train}}|} \sum_{\mathbf{x}\in\mathcal{X}_{\text{train}}} \mathbf{x} \qquad \text{and} \tag{4.4}$$

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{|\mathcal{X}_{\text{train}}|} \sum_{\mathbf{x}\in\mathcal{X}_{\text{train}}} (\mathbf{x}-\hat{\boldsymbol{\mu}})(\mathbf{x}-\hat{\boldsymbol{\mu}})^T \quad . \tag{4.5}$$

In the AD tasks discussed here, the size of the training set $|\mathcal{X}_{\text{train}}|$ is often small compared to the number of features d. In most of our experiments, the training set has no more than 200 images, and d is between 100 and 300 (up to ≈1000 in some cases). This leads to an ill-conditioned covariance matrix $\hat{\boldsymbol{\Sigma}}$, so we use the regularized covariance matrix $\hat{\boldsymbol{\Sigma}}_{\text{s}}$ computed with

shrinkage [Ledoit, 2003; Ledoit, 2004]:

$$\hat{\mathbf{\Sigma}}_{\mathrm{s}} = (1-\alpha)\,\hat{\mathbf{\Sigma}} \;+\; \alpha\,\frac{\mathrm{tr}\left(\hat{\mathbf{\Sigma}}\right)}{d}\,\mathbf{I}_d \quad , \tag{4.6}$$

where $\alpha \in [0, 1]$ is the regularization parameter, $\mathrm{tr}(\cdot)$ is the trace operator, and $\mathbf{I}_d$ is the identity matrix of size d. We use LeDoit-Wolf's method [Ledoit, 2004] to choose α. This estimator ensures a positive definite inverse covariance matrix $\hat{\mathbf{\Sigma}}_{\mathrm{s}}^{-1}$ with optimal regularization parameter based on the number of observations and features in the dataset, achieving a balance between bias and variance.

Inference and Evaluation At inference time, $\hat{\boldsymbol{\mu}}$ and $\hat{\mathbf{\Sigma}}_{\mathrm{s}}$ are plugged into Eq. (4.3), and the anomaly score is used to make a binary decision ("is the image normal or anomalous?") based on a threshold. In this work, we use AUROC as performance metric at test time because it is threshold selection-free. Therefore we do not focus on the operational threshold, but rather on the model's ability to discriminate between normal and anomalous images in the feature space.

Notation For the sake of simplicity, we overcharge the notation and refer to the model's *trained* parameters with $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$ in the following sections. However, recall that, in practice, these parameters are the empirical counterparts $\hat{\boldsymbol{\mu}}$ and $\hat{\mathbf{\Sigma}}_{\mathrm{s}}$ defined above.

1.2 Dimension Reduction

The formulation proposed in the previous section was used by [Rippel, 2021c; Rippel, 2021a] to train multiple MVGs from different layers of the same CNN to improve the detection performance. They also combined principal component analysis (PCA) to truncate the empirical covariance matrix $\mathbf{\Sigma}$ and introduced its variant NPCA. While PCA consists of retaining principal components with the largest variance, NPCA retains those with the smallest variance. They empirically observed that NPCA is systematically more effective for anomaly detection than PCA, which is not intuitive because the opposite is expected for classification tasks.

Other authors [Kim, 2021; Lin, 2022] proposed derivations from their work. [Lin, 2022] theorized that, in the spectrum of principal components, those with more or less variance tend to encode different types of information – some being more relevant for anomaly detection. [Kim, 2021] made similar observations (*i.e.* PCA *vs.* NPCA) with an extension of PaDiM [Defard, 2021] – a pixel position-dependent patch-wise model (*cf.* Fig. 4.1c). On top of that, their results suggested that random projections (as opposed to projecting onto the principal components) is also effective for AD.

Our work is motivated by these observations, aiming to understand their underlying reasons. To approach this problem, we propose a framework that allows for arbitrary combinations of eigencomponents, so (N)PCA can be seen as special cases (Fig. 4.4). Then we introduce an algorithm to optimially select a subset of eigencomponents from $\mathbf{\Sigma}$. In a first moment this framework is tested on a hypothetical scenario. Our results suggest that, in fact, there is no intrisic link between variance (in a projection of the feature space) and the AD performance.

Finally, we re-use the same framework combined with synthetic data to build a fully unsupervised model. We show that our training pipeline consistently improves the anomaly detection performance of the MVG model.

2 Analysis and Optimization via Eigencomponent Selection

First, in Section 2.1, we introduce a framework to optimially truncate the information in $\mathbf{\Sigma}$ by selecting a subset of its eigencomponents, which we coined greedy eigencomponent selection (GreedyES). Then, in Sections 2.2 and 2.3, we present extensive experiements on hypothetical scenarios to anyalize its potential in the context of anomaly detection and how well it can generalize to unseen data. Finally, in Section 2.4, GreedyES is combined with synthetic data to build a fully unsupervised pipeline to enhance the anomaly detection performance of the MVG model.

2.1 Greedy Eigencomponent Selection (GreedyES)

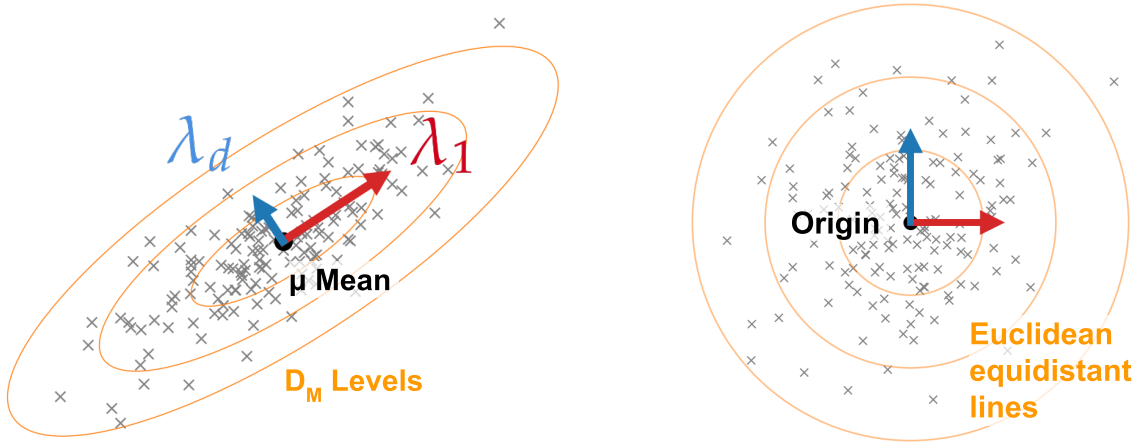
First, we show that the Mahalanobis distance score used in the MVG model can be equivalently computed with the Euclidean norm in a whitened feature space. Adding this whitening operation as an intermediate step simplifies the truncation of eigencomponents of $\mathbf{\Sigma}$ to simply selecting a subset of axes in the whitened feature space. Then, we propose a greedy algorithm to optimize the selection of eigencomponents of $\mathbf{\Sigma}$. Finally, we compare previous works under this framework showing that they are special cases of our formulation.

2.1.1 Whitening

Since $\mathbf{\Sigma}$ is a real symmetric matrix, it can be decomposed as $\mathbf{\Sigma} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$, where $\mathbf{Q}$ is an orthogonal matrix with column $\mathbf{q}_i$ being the i-th eigenvector of $\mathbf{\Sigma}$ and $\mathbf{\Lambda}$ is the diagonal matrix with the element $\mathbf{\Lambda}_{ii} = \lambda_i$ being the i-th eigenvalue of $\mathbf{\Sigma}$. We assume that the eigenvalues are sorted in increasing order, and their eigenvectors are sorted accordingly. The shrinkage regularization (*cf.* previous sections) of $\mathbf{\Sigma}$ ensures that its eigenvalues are real and positive, so a whitening matrix can be built with $\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{Q}^T$. Note that the inverse square root is taken elementwise because $\mathbf{\Lambda}$ is diagonal. Let the *whitened* feature vector $\mathbf{w} \in \mathbb{R}^d$ be defined as

$$\mathbf{w} = \mathbf{\Sigma}^{-\frac{1}{2}}(\mathbf{x} - \boldsymbol{\mu}) = \left(\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{Q}^T\right)(\mathbf{x} - \boldsymbol{\mu}) \quad , \tag{4.7}$$

where $\mathbf{\Sigma}^{-\frac{1}{2}}$ is the square root matrix of the precision matrix $\mathbf{\Sigma}^{-1}$. This affine transformation projects the feature vector $\mathbf{x}$ onto the eigenvectors of $\mathbf{\Sigma}$ and scales it by the inverse square root of the eigenvalues. In other words, $\mathbf{w}$ contains the same information as $\mathbf{x}$ but in a space where the axes are uncorrelated and have unit variance as "seen from the perspective of the normal class". In this space, the anomaly score $\text{MahaDist}_l(\mathbf{x})$ can be equivalently computed with the Euclidean norm of the whitened feature vector $\| \mathbf{w} \|_2$ (*cf.* Fig. 4.3). First, notice that



(a) Original feature space.

(b) Whitened feature space.

Figure 4.3: Whitening operation on the feature space.

the whitening matrix used above can be squared to obtain the precision matrix $\mathbf{\Sigma}^{-1}$:

$$\left(\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{Q}^T\right)^2 = \left(\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{Q}^T\right)\left(\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{Q}^T\right) \tag{4.8a}$$

$$= \mathbf{Q}\left(\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{\Lambda}^{-\frac{1}{2}}\right)\mathbf{Q}^T \tag{4.8b}$$

$$= \mathbf{Q}\mathbf{\Lambda}^{-1}\mathbf{Q}^T \tag{4.8c}$$

$$= \mathbf{\Sigma}^{-1} \tag{4.8d}$$

$$\text{thus} \quad \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{Q}^T = (\mathbf{\Sigma}^{-1})^{\frac{1}{2}} \tag{4.8e}$$

where $(\mathbf{\Sigma}^{-1})^{\frac{1}{2}}$ is the square root matrix of $\mathbf{\Sigma}^{-1}$ (not to be confused with the elementwise square root). With this equality, $\text{MahaDist}_l(\mathbf{x})^2$ (Eq. (4.3)) can be expressed as

$$\text{MahaDist}_l(\mathbf{x})^2 = (\mathbf{x}-\boldsymbol{\mu})^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}) \tag{4.9a}$$

$$= (\mathbf{x}-\boldsymbol{\mu})^T\left[(\mathbf{\Sigma}^{-1})^{\frac{1}{2}}(\mathbf{\Sigma}^{-1})^{\frac{1}{2}}\right](\mathbf{x}-\boldsymbol{\mu}) \tag{4.9b}$$

$$= \left[(\mathbf{\Sigma}^{-1})^{\frac{1}{2}}(\mathbf{x}-\boldsymbol{\mu})\right]^T\left[(\mathbf{\Sigma}^{-1})^{\frac{1}{2}}(\mathbf{x}-\boldsymbol{\mu})\right] \tag{4.9c}$$

$$= \left[\left(\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{Q}^T\right)(\mathbf{x}-\boldsymbol{\mu})\right]^T\left[\left(\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{Q}^T\right)(\mathbf{x}-\boldsymbol{\mu})\right] \tag{4.9d}$$

$$= \mathbf{w}^T\mathbf{w} \tag{4.9e}$$

$$= \|\,\mathbf{w}\,\|_2^2 \tag{4.9f}$$

$$\text{thus} \quad \text{MahaDist}_l(\mathbf{x}) = \|\,\mathbf{w}\,\|_2 \quad . \tag{4.9g}$$

Notice that $\mathbf{w}_i$ – the i-th row of the white vector $\mathbf{w}$ – corresponds to the projection of the centered feature vector $(\mathbf{x}-\boldsymbol{\mu})$ onto the eigenvector $\mathbf{q}_i$, which is normalized by $\lambda_i^{-\frac{1}{2}}$. We introduce this whitening transform in our formulation to simplify the eigendecomposition-based dimension reduction explained in the following section. Selecting a subset of entries from $\mathbf{w}$ corresponds to truncating $\mathbf{\Sigma}$ with a subset of eigencomponents.

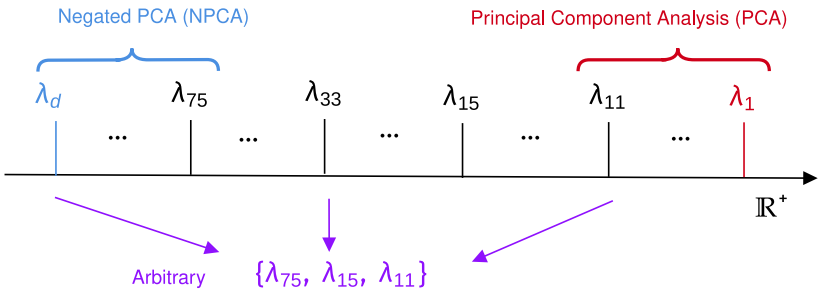


Figure 4.4: Different eigencomponent selections

2.1.2 Greedy Algorithm

We introduce a framework to select eigencomponents of $\mathbf{\Sigma}$, which is equivalent to selecting a subset of axes in the whitened feature space. Let $\mathcal{S}_k$ be the set of subsets of $\mathcal{Q}$ containing k eigenvectors, where $\mathcal{Q} = \{\mathbf{q}_1, \ldots, \mathbf{q}_d\}$ is the set of eigenvectors of $\mathbf{\Sigma}$. Their respective eigenvalues are assumed to be sorted in increasing order. Given a subset $\mathcal{Q}_k \in \mathcal{S}_k$, the covariance matrix $\mathbf{\Sigma}$ is truncated[1] to only include the eigenvectors in $\mathcal{Q}_k$. Their eigenvalues are omitted in the notation for simplicity, but it should be understood that they are selected/excluded along with their corresponding eigenvectors.

Let g be a heuristic that evaluates (higher is better) how well the MVG model (trained on $\mathcal{I}_{\text{train}}$) can detect anomalies on a set $\mathcal{I}_{\text{opt}}$ when truncated by $\mathcal{Q}_k$. For instance, *e.g.* AUROC is used in some of our experiments. Our framework consists of finding the optimal subset $\mathcal{Q}_k^* \in \mathcal{S}_k$ that maximizes g on $\mathcal{I}_{\text{opt}}$:

$$\mathcal{Q}_k^* = \underset{\mathcal{Q}_k \in \mathcal{S}_k}{\operatorname{argmax}} \; g(\mathcal{Q}_k, \mathcal{I}_{\text{opt}}) \quad . \tag{4.10}$$

This optimization allows for arbitrary combinations of eigencomponents, which is not possible with (N)PCA's constraints (Fig. 4.4). The search space in Eq. (4.10) is combinatorial with size $\binom{n}{k}$ (*i.e.* "d choose k") (order does not matter and repetitions are not allowed), which is infeasible to explore exhaustively. To make this problem amenable, we propose to approximate it with a greedy algorithm by iteratively building $\mathcal{Q}_k^*$ one eigencomponent at a time while optimizing g at each step. This is equivalent to Sequential Feature Selection (SFS) [Ferri, 1994a] considering the space of white vectors $\mathbf{w}$ as the feature space.

The GreedyES can be carried out in two ways: by adding or by removing eigencomponents. In the additive case, which we call the "bottom-up" variant (Algorithm 1, illustrated in Fig. 4.5), the algorithm starts with an empty set $\mathcal{Q}_0^* = \emptyset$ then adds eigencomponents one by one such that

$$\mathcal{Q}_i^* = \mathcal{Q}_{i-1}^* \cup \left\{ \underset{\mathbf{q} \in \mathcal{Q} \setminus \mathcal{Q}_{i-1}^*}{\operatorname{argmax}} \; g\left(\mathcal{Q}_{i-1}^* \cup \{\mathbf{q}\}, \mathcal{I}_{\text{opt}}\right) \right\} \quad . \tag{4.11}$$

The top-down variant (Algorithm 2) starts with the full set of eigencomponents $\mathcal{Q}_d^* = \mathcal{Q}$ and

[1] In practice, we include/exclude axes from the whitened features space.

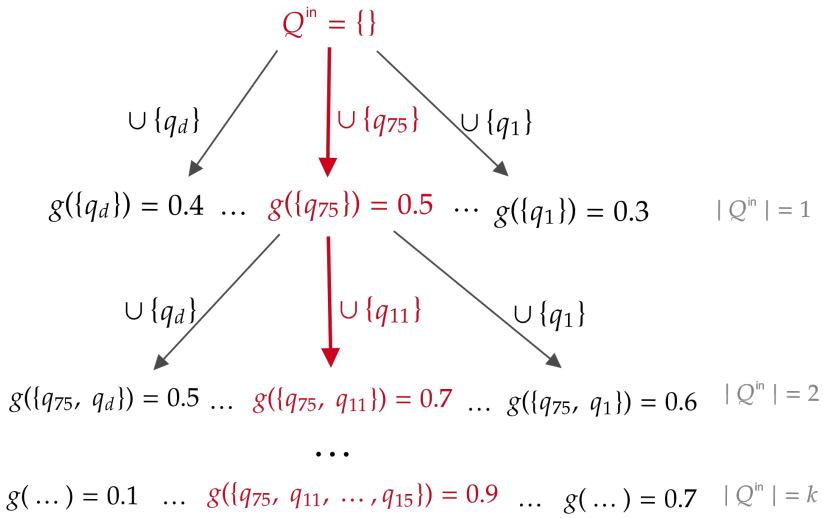


Figure 4.5: Greedy Bottom Up algorithm as tree search. Starting with an empty set, the algorithm iteratively adds the component with the best performance (function g) when combined withe the current selection until it reaches the size k. The red path represents the chosen path at each step. $\mathcal{I}_{\text{opt}}$ is omitted from the notation for clarity.

Algorithm 1 Greedy Bottom Up

Require: $d = |\mathcal{Q}| > 0$ and $1 \leq k \leq d$

procedure GreedyBottomUp($\mathcal{Q}$, k, g)
  $\mathcal{Q}^{\text{in}} \leftarrow \emptyset$ # set of eigenvectors *IN* the model
  $\mathcal{Q}^{\text{out}} \leftarrow \mathcal{Q}$ # set of eigenvectors *OUT* of the model
  while $|\mathcal{Q}^{\text{in}}| < k$ **do**
    $\mathbf{q}^* \leftarrow \text{argmax}_{\mathbf{q} \in \mathcal{Q}^{\text{out}}}\ g\left(\mathcal{Q}^{\text{in}} \cup \{\mathbf{q}\}\right)$
    $\mathcal{Q}^{\text{in}} \leftarrow \mathcal{Q}^{\text{in}} \cup \{\mathbf{q}^*\}$
    $\mathcal{Q}^{\text{out}} \leftarrow \mathcal{Q}^{\text{out}} \setminus \{\mathbf{q}^*\}$

removes them one by one such that

$$\mathcal{Q}_i^* = \mathcal{Q}_{i+1}^* \setminus \left\{ \underset{\mathbf{q} \in \mathcal{Q}_{i+1}^*}{\text{argmax}}\ g\left(\mathcal{Q}_{i-1}^* \setminus \{\mathbf{q}\}, \mathcal{I}_{\text{opt}}\right) \right\} . \tag{4.12}$$

Note that bottom-up is equivalent to forward-SFS and top-down is equivalent to backward-SFS in the context of Sequential Feature Selection (SFS). In practice, Eq. (4.7) makes it simple to simulate subsets of eigencomponents by zeroing non-selected components in the entries of $\mathbf{w}$.

GreedyES should *not* be confused with incremental principal component analysis (IPCA). While IPCA is useful for reducing computational complexity in processing large datasets, our approach is designed to excel at highlighting anomalous images by selecting eigencomponents of $\boldsymbol{\Sigma}$ (all of them are computed).

Algorithm 2 Greedy Top Down

Require: $d = |\mathcal{Q}| > 0$ and $1 \leq k \leq d$

procedure GreedyTopDown($\mathcal{Q}$, k, g)
  $\mathcal{Q}^{\text{in}} \leftarrow \mathcal{Q}$ # set of eigenvectors *IN* the model
  $\mathcal{Q}^{\text{out}} \leftarrow \emptyset$ # set of eigenvectors *OUT* of the model
  while $|\mathcal{Q}^{\text{in}}| > k$ **do**
    $\mathbf{q}^* \leftarrow \text{argmax}_{\mathbf{q} \in \mathcal{Q}^{\text{in}}} \; g\left(\mathcal{Q}^{\text{in}} \setminus \{\mathbf{q}\}\right)$
    $\mathcal{Q}^{\text{in}} \leftarrow \mathcal{Q}^{\text{in}} \setminus \{\mathbf{q}^*\}$
    $\mathcal{Q}^{\text{out}} \leftarrow \mathcal{Q}^{\text{out}} \cup \{\mathbf{q}^*\}$

2.1.3 Previous Works

The eigendecomposition of $\mathbf{\Sigma}$ combined with the whitening operation (and the truncation of eigencomponents) is equivalent to fitting a PCA on $\mathcal{X}_{\text{train}}$. The eigenvectors are the orthogonal directions of maximum variance in $\mathcal{X}_{\text{train}}$, and their eigenvalues correspond to the standard deviations of the data projected onto these directions.

Using our formulation, a k-dimension reduction with NPCA corresponds to replacing $\mathbf{w}$ with $\mathbf{w}_{1:k} = (\mathbf{w}_1, \ldots, \mathbf{w}_k)^T$, and PCA corresponds to replacing it with $\mathbf{w}_{d-k+1:d} = (\mathbf{w}_{d-k+1}, \ldots, \mathbf{w}_d)^T$ (*cf.* Fig. 4.4). This is a more restricted version of our framework, where the choice of $\mathbf{w}_i$ is constrained to adjacent eigencomponents from either the smallest or the largest part of the spectrum. Our results show that these constraints inhibit the potential of dimensionality reduction to improve the anomaly detection performance.

Besides, the parametrization of (N)PCA is commonly expressed by the ratio of explained variance (by the k selected components), which can be expressed as

$$\frac{\sum_{i=1}^{k} \lambda_i^2}{\sum_{i=1}^{d} \lambda_i^2} \quad . \tag{4.13}$$

While meaningful metric for (N)PCA, this formulation is hard to interpret in our context because the feature spaces generated by ϕ_l concentrate most of the variance in a few components. These components are not necessarily the most informative for AD as shown by previous works and our results. Therefore, we prefer to use the number of retained dimensions k as the parameter for our framework instead of entangling it with the variance, which our results suggest to be a spurious relation.

Previous works have given little attention to the choice of k (or the retained variance), generally using a fixed value or testing a few values. In deep feature selection (DFS) [Lin, 2022], the authors proposed two heuristic methods to select k. The authors proposed to split the eigencomponents in three groups: low variance $\Phi_3 = \{\mathbf{q}_1, \ldots, \mathbf{q}_{k_2}\}$, medium variance $\Phi_2 = \{\mathbf{q}_{k_2+1}, \ldots, \mathbf{q}_{k_1}\}$, and high variance $\Phi_1 = \{\mathbf{q}_{k_1+1}, \ldots, \mathbf{q}_d\}$. The breakpoints k_1 and k_2 are heuristically computed based on the eigenvalues and the rank of $\mathbf{\Sigma}$. For instance, k_2 is computed with

$$k_2 = \underset{k}{\text{argmax}} \; \frac{\lambda_{k+1}}{\lambda_k} \quad . \tag{4.14}$$

The authors hypothesize that these groups tend to capture different aspects of the data, and

empirically test them on image AD tasks. They find that combining the low and medium variance groups ($\Phi_{2,3} = \Phi_3 \cup \Phi_2$), which is equivalent of selecting a heuristic value for k with NPCA, yields the best performance overall.

Despite being inspired by a dimension reduction technique, their work translates to selecting the hyperparameter k that optimizes the anomaly detection performance. The goal is to increase the contrast between the normal and anomalous samples in terms of anomaly scores, like a model selection step in the training process. Our work is aligned with this idea, but we extend the scope of hyper-parameter search. Instead of considering a few heuristic values, we thoroughly evaluate all possible values of $k \in \{1, \ldots, d\}$.

Other MVG-based works have used randomized dimension reduction techniques. PaDiM [Defard, 2021] and Semi-orthogonal [Kim, 2021] used MVG models on patch-wise feature vectors: instead applying a spatial aggregation on the feature maps, each vector in it – corresponding to a patch of the image – is used as a feature vector (ϕ_l as in Fig. 4.1c). PaDiM uses a random feature selection directly on the feature space of ϕ_l, which is equivalent to cutting rows from $\mathbf{x}$ and their respective rows and columns from $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. Semi-orthogonal generalized this idea using random orthogonal projection matrices. The former can be seen as a special case of the latter by using only binary entries in the projection matrix. These two are faster to compute because they do not require decomposing $\boldsymbol{\Sigma}$, and they can be written with an equivalent affine transform like Eq. (4.7).

The authors of Semi-orthogonal base their work on the assumption that the eigencomponents with the smallest eigenvalues are the most informative for AD. However, our results contradict this assumption, suggesting the variance-performance entanglement is not relevant.

Finally, it is worth noting that PatchCore [Roth, 2022] – still SOTA on MVTec-AD (*cf.* the benchmark results in Chapter 2) – also uses dimension reduction internally. Random linear projections are used as an intermediate step to reduce the execution time of the coreset algorithm. However, as the reduced memory bank comes out of it, the dimension reduction is disregarded at inference time. Combining PatchCore with our framework could be a promising direction for future works.

As shown by these previous works, dimension reduction is a promising source of exploration in the context of AD, especially with MVG models. Recent works have focused on engineering ever-more complex training processes, but they mostly end up creating more complicated models with the same principles at their core (*e.g.* probability density estimation, reconstruction error, feature vector error, etc.; *cf.* Chapter 1). We believe that understanding the feature space of neural networks – in this case, through the lens of MVG models and their decomposition – is a promising direction that has been overlooked. Our work intends to provide insights about the feature space itself and the underlying structure of the data.

2.2 Overfitting and Generalizing with GreedyES

In Section 2.2.1 we establish a general setup to study GreedyES with different data splits for the execution of the greedy optimization and its evaluation. Then, in Section 2.2.2 we seek to understand the potential of our approach algorithm. Finally, in Sections 2.2.3 and 2.2.4 we focus

on understanding its generalization capabilities.

It should be emphasized that the experiments in Sections 2.2.2 to 2.2.4 focus on observing the behavior of GreedyES *across datasets and layers* – *not* on model selection. We do *not* focus on cross-category average performance, but rather *per*-category performances because they are all independent AD tasks. While the former provides a literature-comparable summary metric, our focus is to identify phenomena that generalize across different data. We show representative cases here for brevity, but the generality of our observations can be seen in Appendices 2 to 4 (extracted from [Gula, 2023]), where all scenarios are documented.

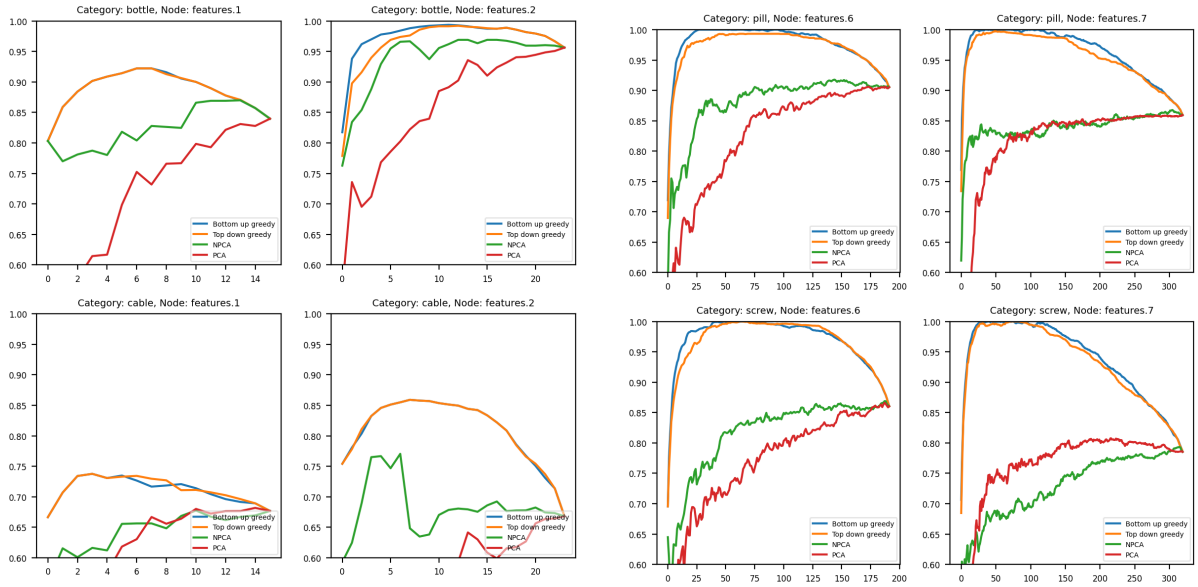
2.2.1 Setup

GreedyES is analyzed across the major blocks of layers[2]of an EfficientNet backbone and across all categories in MVTec-AD. In a given scenario (fixed category and layer), all the possible values of $k \in \{1, \ldots, d\}$ are evaluated, and results are generally visualized as k-vs-AUROC curves. In these curves, the reduced number of dimensions k is on the X-axis and its respective (image-wise) AUROC on the evaluation data split on the Y-axis. Notice that the baseline (no dimensionality reduction) corresponds to the right-most point on these curves, where $k = d$. Finally, notice as well that d is different for each layer, so the X-axis is not directly comparable across layers.

Performance Metric We use AUROC [Fawcett, 2006] both as the g function and as evaluation metric. Here we use the *image-level* AUROC, which should not be confused with the pixel-level AUROC discussed in Chapter 2. The AUROC score is a widely used metric for AD and conveniently threshold selection-free. It measures the probability that the anomaly score assigned to an anomalous instance is higher than that assigned to a normal instance. An AUROC of 0.5 indicates random guessing, and an AUROC of 1 indicates perfect discrimination between normal and anomalous instances.

Dataset and Data Split We use the datasets from the MVTec Anomaly Detection (MVTec-AD) collection [Bergmann, 2019; Bergmann, 2021a] (see Fig. 1.1) for our experiments. The normal samples from the train set are used to fit the MVG model $\mathcal{N}(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}_{\mathrm{s}})$. The test set is divided into optimization split $\mathcal{I}_{\mathrm{opt}}$ and evaluation split $\mathcal{I}_{\mathrm{eval}}$. The split $\mathcal{I}_{\mathrm{opt}}$ is used to execute GreedyES, in the function g (Algorithm 1 and Algorithm 2), and the set $\mathcal{I}_{\mathrm{eval}}$ is used to evaluate the models (reported values in the results). We do this because the g function is evaluated on independent samples then evaluated on unseen data – however, Experiment 1 (Section 2.2.2) uses the same split for both on purpose. Both splits contain normal and anomalous instances, which is necessary for the performance metric computation. A fully unsupervised scenario should assume no access to anomalous samples, but our focus is to gain insights into the MVG model. A no-access-to-anomalies scenario is presented in Section 2.4, where synthetic anomalies are used as a proxy for the true anomalies.

[2]The notion of "blocks of layers" here is based on the notion of "nodes" in `pytorch`, and their names come from the `torchvision` implementation of EfficientNet.



(a) {bottle, cable} × {`features.1`, `features.2`} (b) {pill, screw} × {`features.6`, `features.7`}

Figure 4.6: Experiment 1: selection of representative k-vs-AUROC curves. X-axis: number of eigencomponents selected (k). Y-axis: AUROC on the evaluation set ($\mathcal{I}_{\text{eval}} = \mathcal{I}_{\text{opt}}$ here). A complete report (all categories and nodes) is available in Appendix 2, extracted from [Gula, 2023].

Feature Extractor Like DFS [Lin, 2022], EfficientNet [Tan, 2019] is used as backbone for the feature extraction. Specifically, we use EfficientNetB0 pre-trained on classification on ImageNet[3]. We analyze the nine main nodes from EfficientNetB0, which are sequentially named from "`features.0`" to "`features.8`"[4]. The vectors' size d usually ranges from 10s to 100s, reaching up to nearly 1000 in the deepest node (`features.8`). Our backbone choice is mostly based on previous works for comparability. We train models using different layers from the same backbone, so we can observe how the shallow and deep layers behave differently. As EfficientNets has many (nine) major nodes, the transition of behavior happens more smoothly than, for instance, ResNet [Liu, 2020] (four major blocks).

2.2.2 Experiment 1: Overfit

We set $\mathcal{I}_{\text{opt}} = \mathcal{I}_{\text{eval}} = \mathcal{I}_{\text{test}}$ to intentionally overfit the evaluation set. The goal here is *not* to learn, but rather analyze the performance improvement that could be achieved to validate the potential of GreedyES. Despite this setup being unrealistic, it is useful for diagnostic purpose because one can measure GreedyES's *potential* compared to PCA, NPCA [Rippel, 2021c], and DFS [Lin, 2022]. It also shows the impact of the constraints imposed by (N)PCA – *i.e.* eigencomponent choice based on their variances λ_i^2. Furthermore, additional analyses also reveal interesting insights into the feature extractor itself.

In Experiment 2 (Section 2.2.3) and Experiment 3 (Section 2.2.4), $\mathcal{I}_{\text{opt}}$ and $\mathcal{I}_{\text{eval}}$ do *not* have

[3] EfficientNet_B0_Weights.IMAGENET1K_V1 from `torchvision`.
[4] `torchvision`'s notation for the nodes' names in EfficientNet.

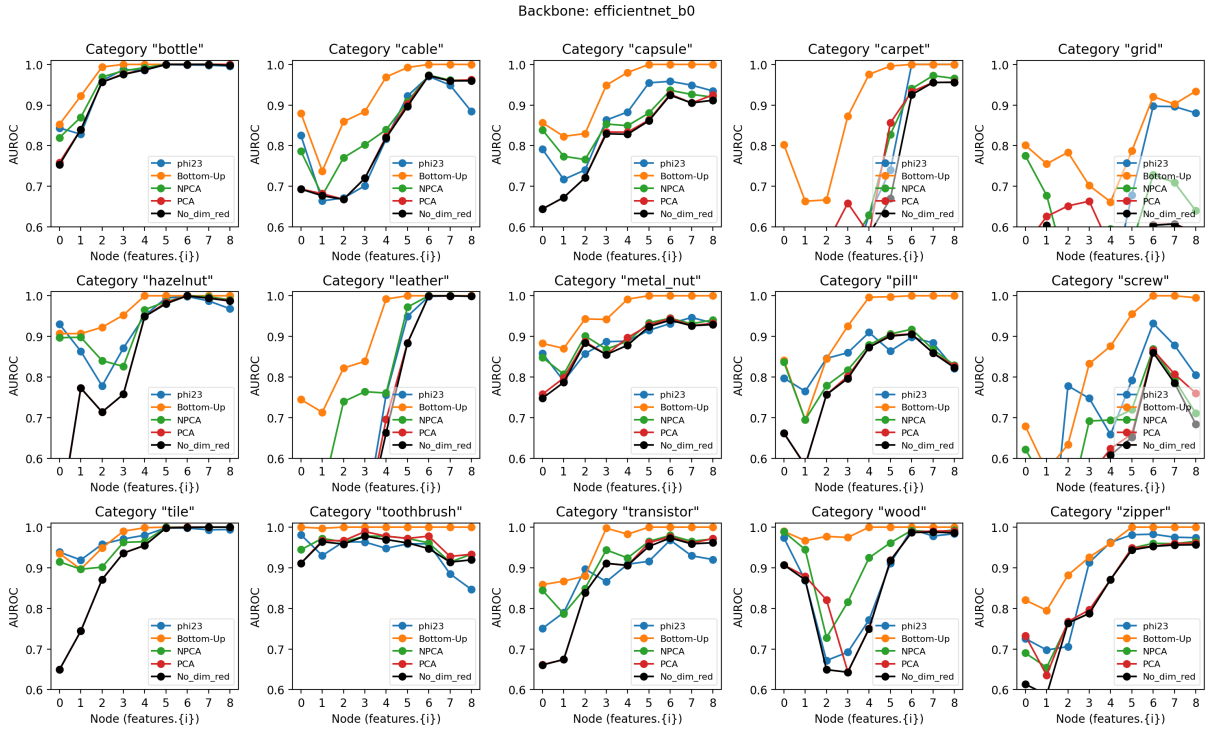


Figure 4.7: Experiment 1: best AUROC out of all values of k per node. X-axis: index of the node in the backbone. Y-axis: best AUROC achieved by that node (pick of the curves in Fig. 4.6). The curve "phi23" refers to the results from [Lin, 2022] with the alternative "$[\Phi_2, \Phi_3]$".

anomalous images in common so the generalization power of our proposed framework can be compared to the *achievable* performances revealed by Experiment 1. In particular, we focus Experiment 2 and Experiment 3 on the nodes from `features.5` up to `features.8` because they can reach 100% AUROC and show more stable behavior.

Results Fig. 4.6 shows a selection of k-vs-AUROC curves from representative cases in Experiment 1. GreedyES (bottom-up and top-down) is compared with PCA and NPCA. Notice that at the point where $k = d$ (*i.e.* no dimension reduction) all the lines merge because they simplify to the same model. Fig. 4.7 shows summary plots from each category. The X-axis is the node index (higher values mean "deeper in the network") and the Y-axis is the best AUROC achieved by that node. Each point corresponds to extracting the maximum AUROC from a k-vs-AUROC curve (*e.g.* Fig. 4.6). Only bottom-up is shown in Fig. 4.7 because the results with top-down are very similar. Additionally, we compare it with the results achieved in [Lin, 2022][5], noted "phi23" for $\Phi_{2,3}$ (*cf.* Section 2.1.3). We also show the results when no dimension reduction is applied for comparison.

Discussion The analysis exemplified in Fig. 4.6 suggests that it is *possible* to (sometimes greatly) enhance the performance of the MVG model by cherry-picking eigencomponents from $\mathbf{\Sigma}$. As shown in Section 2.3.2, deeper layers require fewer components to achieve high performance

[5]We thank the authors for having shared their results data with us.

– in relative terms ($\frac{k}{d}$) and sometimes even in absolute (k) terms. Most scenarios using bottom-up exhibit a three-phase behavior. First, there is a (near) monotonic increase in performance until reaching a saturation level. Then, the performance remains stable in a plateau for a range of k values. Finally, there is a decrease in performance. This behavior is further detailed in Section 2.3.1, where we split the k-vs-AUROC curves in these three "phases", which we coin as "rise", plateau, and "drop". Some scenarios, however, show an edge case behavior with a rise followed directly by a drop of performance. For instance, categories "metal nut", "pill", and "screw" with `features.8`, and category grid with the last four nodes (*cf*. Fig. A.31).

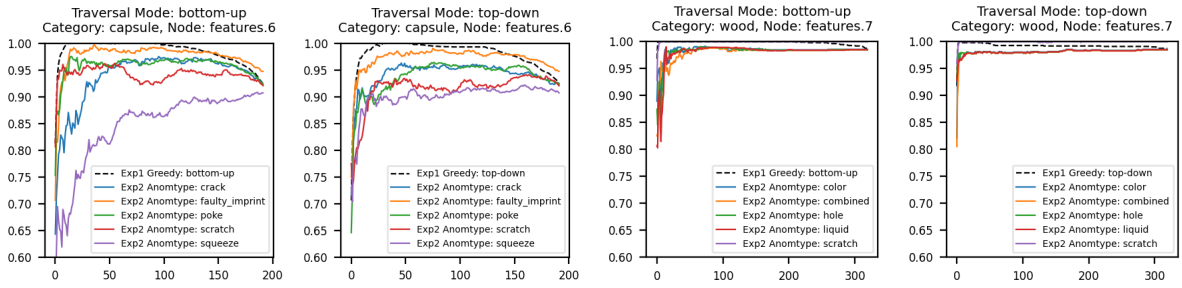
(N)PCA, even at optimal k, are generally bellow the achievable performance demonstrated by GreedyES (see Fig. 4.6). High performance can be achieved using only 30-40 components, and even nearly-perfect class discrimination (100% AUROC) with less than ten components. For instance, see categories "bottle", "carpet", "hazelnut", "leather", "toothbrush", "tile", "wood" in the Section 2.3.1. In other words, it is possible to encode the normality of a semantic class in very small embeddings (compared to d). This underscores the importance of dimension reduction for deep layers of the network, where the number of dimensions can be substantial (100s or even 1000s).

Shallow vs. Deep Layers It has been observed [Rippel, 2021a; Lin, 2022; Defard, 2021; Roth, 2022] that mid-depth (nor shallowest nor deepest) nodes yield best performance. Our results, however, contradict this observation. Fig. 4.7 shows that – except for category "grid" – deeper layers tend to be more informative for AD than shallower ones, being capable of achieving perfect score (or very close to it). When using (N)PCA or no dimension reduction something different happens. The deepest layers tend to show a drop in performance (*e.g.* categories "capsule", "pill", "screw", "toothbrush"), which was believed to be due to a bias towards the pre-training task.

Bottom-Up vs. Top-Down bottom-up and top-down, behave similarly in most scenarios. However, bottom-up tends to achieve better results with longer plateaus, therefore successfully avoiding spurious eigencomponents.

2.2.3 Experiment 2: Generalization with a Single Anomaly Type

Experiment 2 consists of splitting the anomaly types in $\mathcal{I}_{\text{opt}}$ and $\mathcal{I}_{\text{eval}}$. The candidate eigencomponents are ranked (function g) based on a single anomaly type in $\mathcal{I}_{\text{opt}}$. The evaluation split encompasses all the other anomaly types, and both $\mathcal{I}_{\text{opt}}$ and $\mathcal{I}_{\text{eval}}$ use all the normal images from the test set. This experiment enables us to assess how well GreedyES generalizes from a single anomaly type. Fig. 4.8 shows a selection of k-vs-AUROC curves from representative cases in Experiment 2. All the data splits for a given category (named after the anomaly type used in $\mathcal{I}_{\text{opt}}$) are compared with the respective curve from Experiment 1. In both modes (bottom-up and top-down), we observe that GreedyES can reduce the number of dimensions – generally – without losing performance, as it is the case for (N)PCA. However, it often underperforms compared to the best achievable performance in Experiment 1.



(a) capsule, `features.6` (b) wood, `features.7`

Figure 4.8: Experiment 2: selection of representative k-vs-AUROC curves. X-axis: number of eigencomponents selected (k). Y-axis: AUROC on the evaluation set ($\mathcal{I}_{\text{eval}} \neq \mathcal{I}_{\text{opt}}$ here). A complete report (all categories and nodes) is available in Appendix 3, extracted from [Gula, 2023].

While Experiment 1 shows it is possible to obtain a perfect classifier with less than 30 eigencomponents – and adding up to other 70 eigencomponents does not hurt the performance – none of the single-anomaly-type runs were capable of reaching such performance. Most scenarios (*cf.* [Bertoldo, 2023a]) have this behavior with more or less variability across anomaly type. For instance, category "capsule" (Fig. 4.8a) has a stronger dependency on the anomaly type than the category "cable" (Fig. 4.8b). Still, GreedyES runs in Experiment 2 are comparable and more often better than (N)PCA. This comes without surprise due to the supervision used in GreedyES – encoded in g – which is not available in (N)PCA. However, even in such cases, it often fails to achieve the same level of performance seen in Experiment 1.

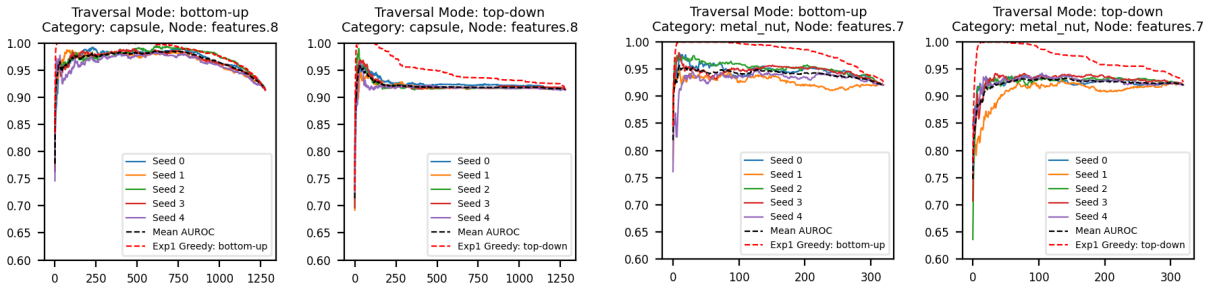
Bottom-Up vs. Top-Down bottom-up often reaches better maximum performance with lower k, while top-down is more stable at keeping the baseline performance (no dimension reduction) and shows a less variable behavior. Categories "carpet" and "zipper" (node `features.7` in particular) are good examples of such contrast. Other examples include categories "hazelnut", "leather", "transistor", and "wood".

2.2.4 Experiment 3: Generalization with Multiple Anomaly Types

Instead of basing the choice of components on the whole test set (Experiment 1) or on a single anomaly type (Experiment 2), we now sample images from all anomaly types. Given a fixed budget of 15 images, we randomly split the data with 5 seeds such that $\mathcal{I}_{\text{opt}}$ has at least 15 anomalous images with equal number of images per anomaly type. The remaining anomalous images go to $\mathcal{I}_{\text{eval}}$. Both $\mathcal{I}_{\text{opt}}$ and $\mathcal{I}_{\text{eval}}$ use all the normal images from $\mathcal{I}_{\text{test}}$.

Fig. 4.9 shows a selection of k-vs-AUROC curves from representative cases in Experiment 3. Each curve corresponds to a seed, and their cross-seed mean curve and the curve from Experiment 1 are plotted together for reference. The results from nodes from `features.0` up to `features.4` have been omitted because their *achievable* performances (see Experiment 1 in Section 2.2.2) are generally worse.

Compared to Experiment 2, generalization is slightly better. Results show less variance



(a) capsule, `features.8` (b) metal nut, `features.7`

Figure 4.9: Experiment 3: selection of representative k-vs-AUROC curves. X-axis: number of eigencomponents selected (k). Y-axis: AUROC on the evaluation set ($\mathcal{I}_{\text{eval}} \neq \mathcal{I}_{\text{opt}}$ here). A complete report (all categories and nodes) is available in Appendix 4, extracted from [Gula, 2023].

across runs of a same scenario, which is expected because $\mathcal{I}_{\text{opt}}$ is not biased towards a single anomaly type. However, two counter examples are worth noting: categories "pill" and "transistor". Experiment 1 revealed a important margin for improvement (higher performance with few components) relative to the baseline (no dimension reduction). Unfornately, Experiment 3 shows that the ability to generalize with reduced amount of data is limited, and the discrepancy is usually bigger for the bottom-up mode. GreedyES fails to avoid bad components, although performances are generally better than (N)PCA, which is not surprising due to the supervision. Noticeable examples include categories "capsule", "carpet", "metal nut", "pill", and "screw".

Fig. 4.9a shows an encouraging example where the bottom-up approach successfully generalizes to unseen data. Most runs achieved substantial performance improvement relative to the baseline (no dimension reduction) with low variability. Category "carpet" with node `features.6` and category "leather" with node `features.5` also represent well such behavior. In cases where the baseline typically has more than 95% AUROC, this is often the case as well. Although the relative improvement is not as prominent (baseline is already high), the dimensionality reduction – at nearly-constant performance – is considerable This reveals that the feature space is redundant for a given dataset. Some examples include categories "cable" and "zipper" with nodes from `features.6` to `features.8`, and categories "carpet" and "hazelnut" with node `features.8`.

Bottom-Up vs. Top-Down The same pattern observed in Experiment 2 is seen here: bottom-up is more embedding-size-efficient, while top-down manages to only keep the baseline performance (no dimension reduction). In Experiment 3, however, the two modes do not differ as much, making the top-down a safer choice. Examples of this can be seen in categories "capsule" and "hazelnut" (nodes from `features.6` to `features.8`).

2.3 Other Analyses

The experiment in Section 2.2.2 shows that the features learned by the CNN ϕ can be used to build surprisingly small (yet effective) embeddings to detect anomalies in images. In Fig. 4.6b, for example, GreedyES reaches 100% AUROC with less than eigencomponents out of (around) 300

while the best (N)PCA reaches less than 85% AUROC. In this section, more detailed analyses of this experiment are presented to provide a better understanding of such – somewhat surprising – results.

In Section 2.3.1, we formalize the notion of "regimes" (or phases) observed in the k-vs-AUROC curves in Experiment 1. A strict criterion is used to identify the rise, plateau, and drop regimes, so the cases that fit this criterion are reused in the following analyses.

In Section 2.3.2, we analyze the minimal number of dimensions required to achieve the best AUROC score at a given node. The results show that deeper nodes – which have more dimensions – require fewer components to achieve the best performance.

In (N)PCA, the ratio of explained variance (Eq. (4.13)) is used as a reference signal to parametrize the dimension reduction. However, Section 2.3.3 shows evidence that this is a bad design choice for AD tasks. Our results reveal that there is no connection between eigencomponent-wise variance λ_i^2 and its AD suitability. Otherwise said, the eigencomponents with the largest or smallest variance do not discriminate normal from anomalous data better than one another, contradicting the core premise of (N)PCA.

Section 2.3.4 shows two simulations investigating the nature of the three regimes. Results suggest that the eigencomponents in the plateau are redundant with the most useful ones, and eigencomponents in the drop regime have "inverted" features. These features provoke performance drops faster than noise, so they likely assign higher activations to normal images. This result is counterintuitive, as the MVG model is expected to do the opposite. However, the visualizations shown in Section 3 [Bertoldo, 2023c] explain this behavior, which is due to memorization of sub-parts of the normal images (*e.g.* image borders are particularly problematic).

2.3.1 Regimes

Fig. 4.10 is extracted from the results of Experiment 1. We distinguish three main regions in the k-vs-AUROC curves from the greedy eigencomponent selection of the bottom-up traversal mode, which we refer to as "regimes". We designate the "Rise" regime in blue, the Plateau regime in red, and the "Drop" regime in green. Formally, the regimes are defined from the minimum and maximum values of k such that the corresponding AUROC is 100%, noted $k_{\text{plat-min}}$ and $k_{\text{plat-max}}$ respectively. Rise is defined as the interval $[1, k_{\text{plat-min}} - 1]$, Plateau as $[k_{\text{plat-min}}, k_{\text{plat-max}}]$, and Drop as $[k_{\text{plat-max}} + 1, d]$.

We selected cases where AUROC reaches the score of 100% at some point in the curve, which is means that normal and anomalous instances are perfectly separated. We deliberately only show deep nodes starting with `features.5` because the regimes are clearly visible for many categories.

The following observations can be made from the Fig. 4.10:

1. Rise (blue): this phase represents the initial selection of eigencomponents where each new addition significantly boosts the performance.

2. Plateau (red): in this phase, the addition of more eigencomponents does not significantly improve or degrade the performance; for most cases, 100% AUROC is achieved at all values

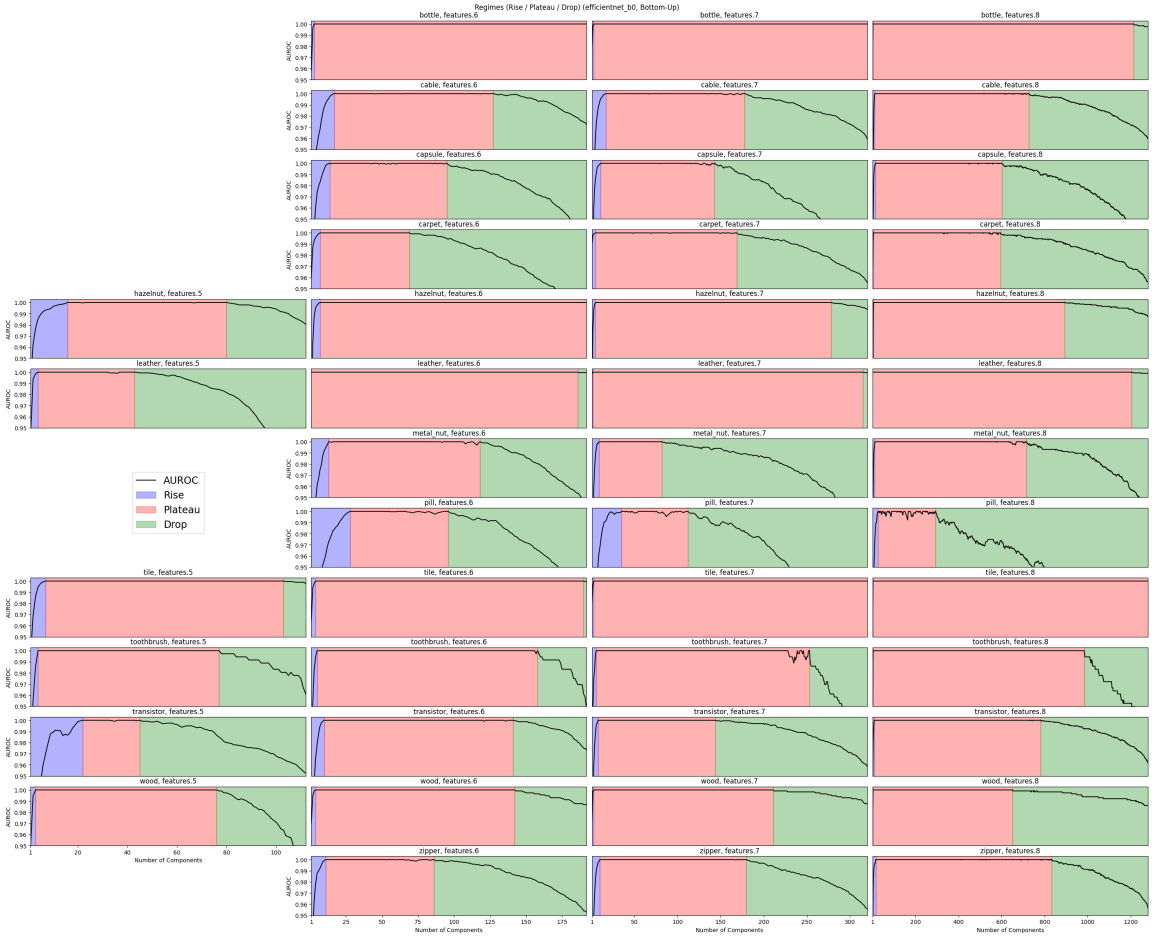


Figure 4.10: Experiment 1: eigencomponents regimes. X-axis: number of eigencomponents selected (k). Y-axis: AUROC on the evaluation set ($\mathcal{I}_{\text{eval}} = \mathcal{I}_{\text{opt}}$ here). Best viewed in color in digital version.

of k within this regime.

3. Drop (green): phase where adding more eigencomponents degrades the performance. This can happen due to the incorporation of noisy, irrelevant, or redundant components.

2.3.2 Minimal Number of Dimensions

Fig. 4.11 shows an analysis of the optimal dimension reduction size for all the scenarios in Experiment 1 (Section 2.2.2). We select minimal number of dimensions k such that its corresponding AUROC is maximal. It corresponds $k_{\text{plat-min}}$ from the Section 2.3.1, which is the left most point of the plateau regime.

The x-axis corresponds to the node depth in EfficientNetB0. On the y-axis, the dots (scaled on the left) shows $k_{\text{plat-min}}$ and the dashed line (scaled on the right) shows the original feature vector size d. The marker color represent a point's corresponding AUROC scaled from 0.90 to 1.00 (light blue to pink).

The optimal number of components (left y-axis) of high-performance models (pink markers) has a negative trend relative to the node depth (deeper nodes implicate less components), while the original embedding size (right y-axis) is bigger. In other words, higher dimensional embeddings tend to be capable of encoding the normality of the images in lower dimensional subspaces.

2.3.3 Low vs. High Variance Components

We analyse if the order that components show up in the bottom-up relates to PCA or NPCA. Fig. 4.12 shows the step index vs. the component index of all the runs with EfficientNetB0 using the bottom-up strategy. The step index in the x-axis corresponds, from left to right, to the depth of the search tree in GreedyES (illustrated in Fig. 4.5) – in other words, the order that eigencomponents were added. The component index in the y-axis correponds to the ordering of the eigenvalues from lowest to highest variance.

PCA's plot would be a line with slope -1 (first component is the largest, last component is the smallest). NPCA's plot would be a line with slope 1 (first component is the smallest, last component is the largest). The scenarios (*i.e.* a category-node pair) present in Section 2.3.1 are colored according to the regimes defined there (*cf.* Fig. 4.10).

Our goal is to understand if the variance of the components is related to the AD performance. These plots give insight into the relation between the variance of the components and their utility for AD. In particular, we are interested in the components that most efficiently distinguish between normal and anomalous instances – thus those added in the beginning of the greedy search. The most relevant nodes are from `features.5` to `features.8` because they achieve the best performances (*cf.* Fig. 4.7). Within those, the most relevant components are generally the first 10 up to 30 ones (left most part of each plot; see Fig. 4.11), and they generally corresponds to the rise regime.

General Behavior Most plots are disordered and seemingly random. Notice how the lack of entanglement between the two is particularly visible at the first (thus most useful) components. This shows that there is no relation between eigenvalue magnitude (*i.e.* the amount of variance encoded in an eigencomponent) and utility for AD. This contradicts several previous works using PCA, NPCA, and in [Lin, 2022].

Exceptions Some cases (*e.g.* some nodes from categories bottle and tile) are strongly ordered like PCA. However this is an artifact due to tie conditions (all components yield the same AUROC, *cf.* Fig. 4.10). Several cases also show a triangular-like shape, which means that the component selection alternates between high and low variance components. However mostly happens in the plateau regime, where the choice of performance is mostly insensitive to adding more components.

Conclusion This analysis explains why (N)PCA require a larger k to achieve their best score and it is yet systematically lower than GreedyES's. The constraint imposed to select eigen-

components based on their variance results in selecting spurious ones from the drop regime (Section 2.3.1). On the other hand, NPCA consistently outperforms PCA, suggesting indeed some (spurious?) correlation between eigencomponent variance and AD performance.

Figure 4.11: Experiment 1: minimal number of (reduced) dimensions k at maximum AUROC. Best viewed in color in digital version.

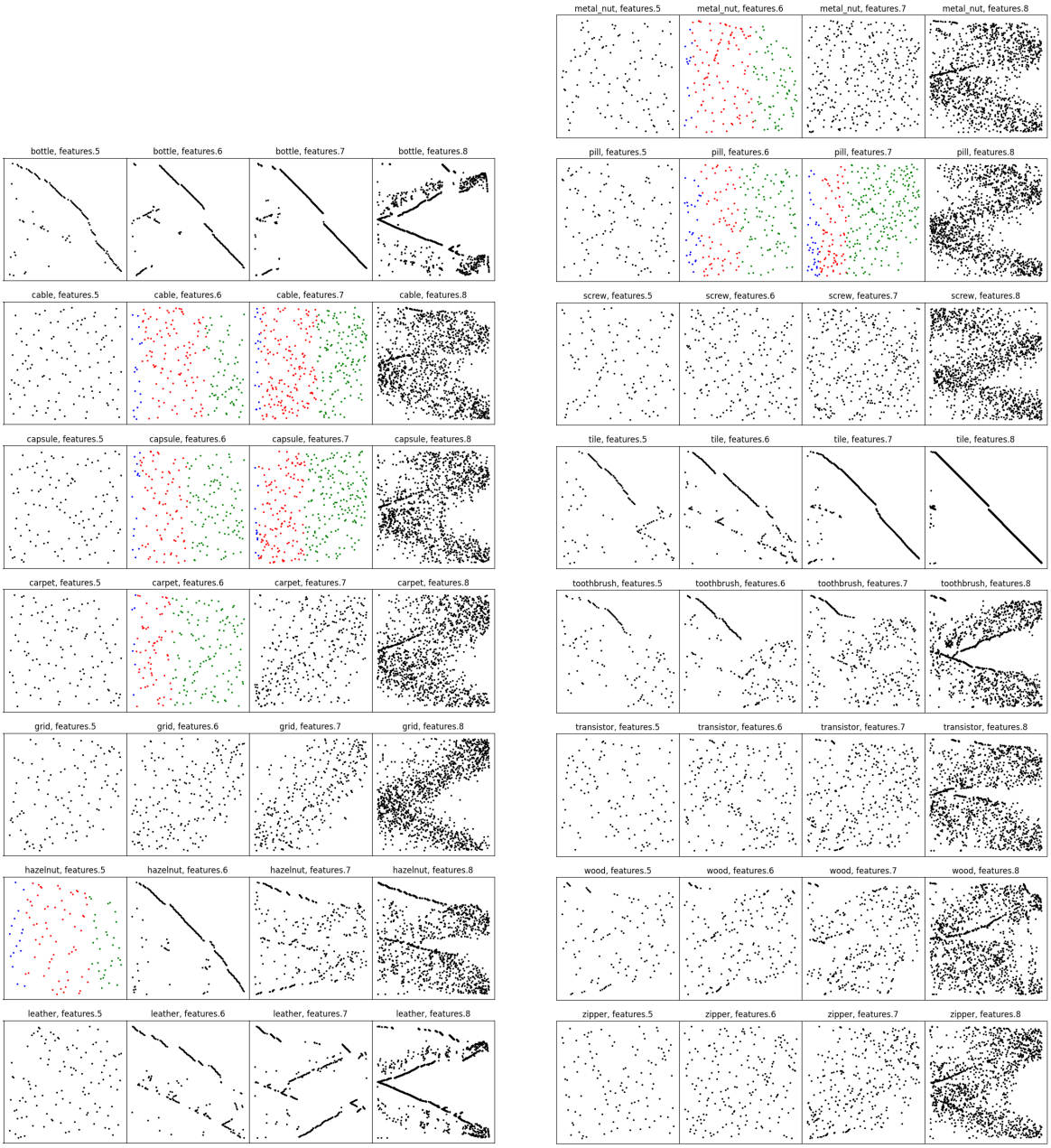


Figure 4.12: Order of eigencomponents in the bottom-up greedy search. The x-axis is the step index, the y-axis is the component index. From left to right, the order that the eigencomponents were added. From bottom to top, the smallest to highest eigenvalues (*i.e.* increasing order of variance). The scenarios presented in Fig. 4.10 are colored accordingly. Best viewed in color in digital version.

2.3.4 Redundant, Noisy, and Spurious Eigencomponents

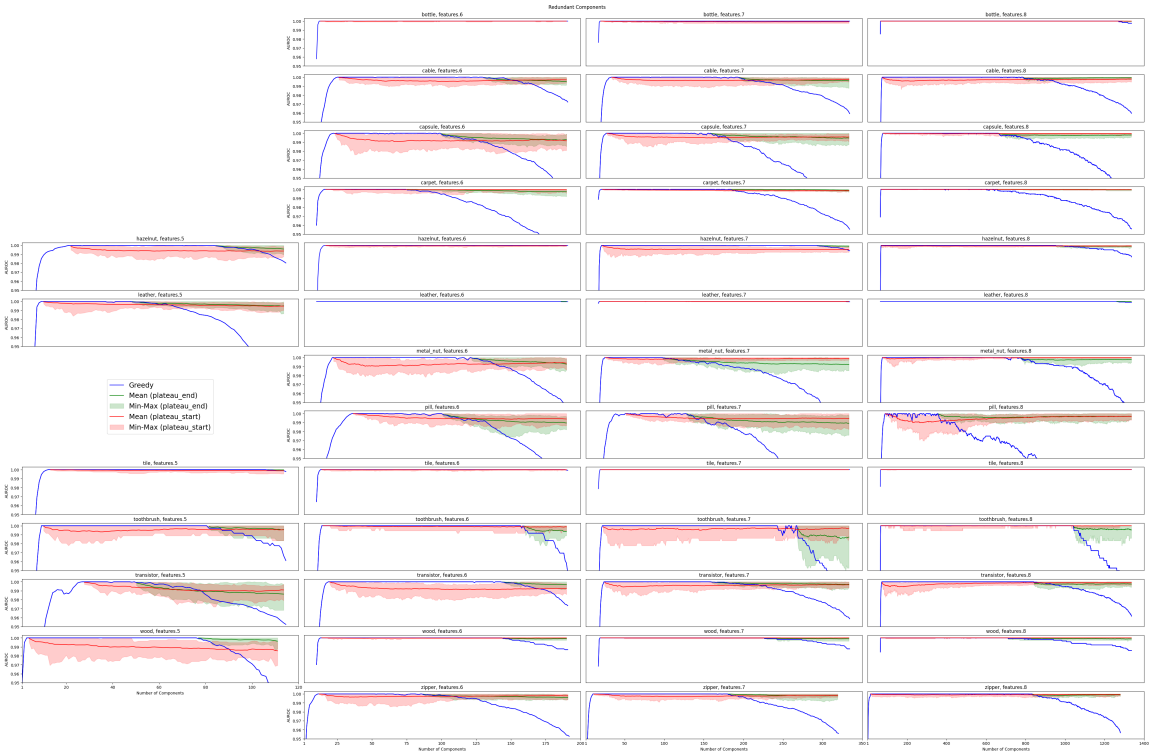
Can one say that the components selected after the rise regime contain noisy or even spurious information? Fig. 4.13b presents the outcomes of a simulation that addresses this question.

The d eigencomponents are sorted as they were yielded by the results of the bottom-up greedy search using the full test set (*i.e.* overfit experiment from Section 2.2.2). A dimension reduction with k components corresponds to the k-th step (or depth, *cf*. Fig. 4.5) of the search tree traversal. This order of eigencomponents is interpreted as "from the most useful to the least useful (or most harmful)" relative to the AD performance.

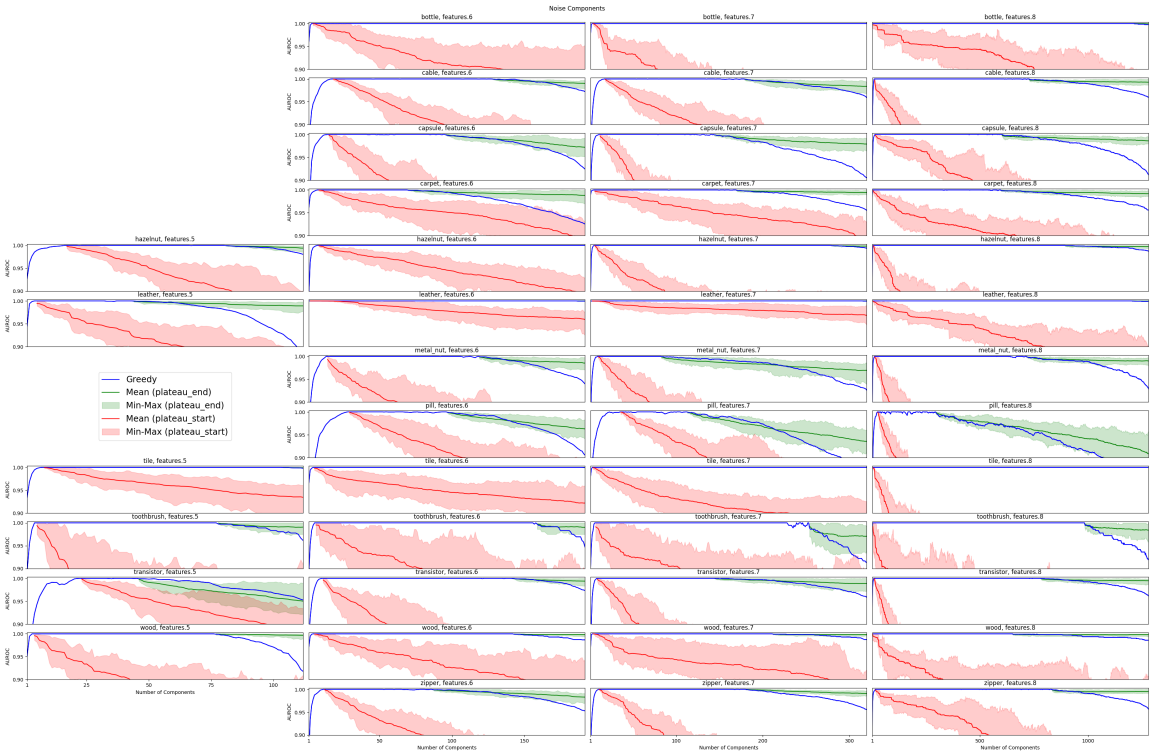
Given a start position k' of this sequence, the eigencomponents before (*i.e.* for $k \leq k'$) are fixed, but the rest (*i.e.* for $k > k'$) is replaced by a synthetic signal. In other words, $\forall k > k'$ the performance is evaluated a combination of the actual eigencomponents up to k' and a synthetic signal. The respective performance of each simulated dimension reduction $k > k'$ is recorded and compared to the the original dimension reduction with the same k. Example: for a feature vector with size $d = 10$, a starting position $k' = 4$, and simulated $k = 6$, the first four eigencomponents are fixed, and the last two are replaced by noise. Two starting positions are considered: $k' \in \{k_{\text{plat-min}}, k_{\text{plat-max}}\}$, which are the first and the last position of the plateau regime. This corresponds to retaining, respectively, the rise components and the rise + plateau components.

Two types of synthetic signals are considered: noise and redundancy. The noise signal is drawn from a normal distribution (each fake component is independent). The redundant signal is a Gaussian random projection of the $k' - 1$ original eigencomponents (*i.e.* multiple random linear combinations of the actual signal). To make a fair comparison, the synthetic signal at position $i \in \{k', \ldots, d\}$ is scaled such that its empirical variance on the test set equals that of the original component i. The four combinations of the aforementioned parameters (noise/redundancy, start position k') were considered tested. Each scenario is repeated with 30 different random seeds. Figs. 4.13a and 4.13b show the minimum, average, and maximum AUROC at each k value.

Most categories and nodes show a consistent behavior: compared to the components in the plateau regime, noise deteriorates the performance, and faster (with less synthetic signals) than the redundant signal. The latter, on the on the other hand, often remains close to the plateau's performance. This suggest that the eigencomponents in the plateau have indeed redundant information. The eigencomponents in the drop regime, however, have spurious features that do not discriminate the normal from the anomalous class – in fact provoking a faster performance drop than pure noise.

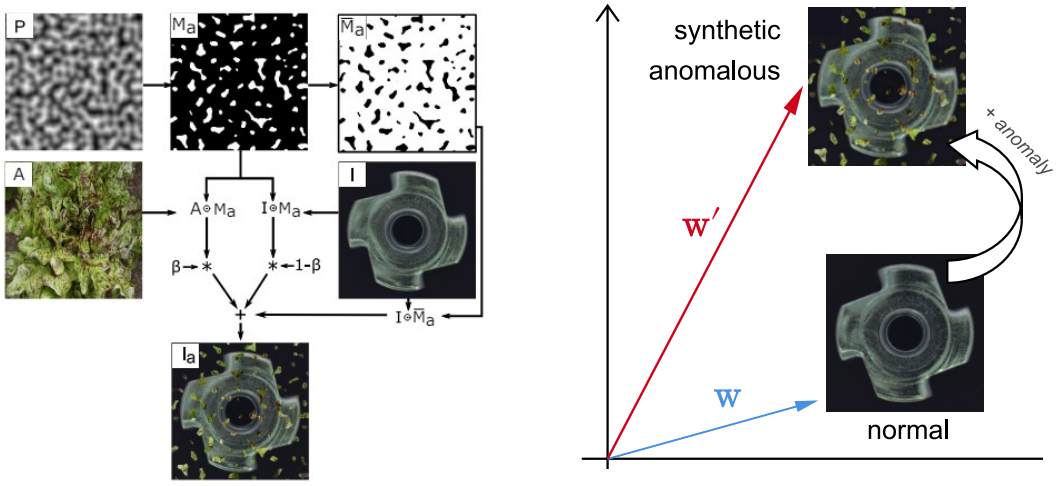


(a) **Redundant** signal.



(b) **Noisy** signal.

Figure 4.13: Simulated performance with synthetic signals. Red/green: replacement starts at the start/end of the plateau regime. Best viewed in color in digital version.



(a) (b)

Figure 4.14: (a) Synthetic anomaly generation scheme from DRAEM [Zavrtanik, 2021a]. Image copied from [Zavrtanik, 2021a]. (b) Intuition of the synthetic anomaly generation: an anomaly detection-effective feature space would push the feature vectors of the synthetic anomalies away from the mean of the normal data (*i.e.* the origin in the whitened space above).

2.4 GreedyES Optimization with Synthetic Anomalies

In Section 2.2, we showed that the MVG model can leverage the activation maps from a backbone CNN to build very AD-efficient embeddings with GreedyES. As a matter of fact, a very small portion of eigencompoent projections is enough to achieve high AUROC scores (*cf*. Fig. 4.11). However, the estimation of this optimal subset of eigencomponents is hard to achieve in practice – especially when the modes of anomalies are not known.

The experiments in Sections 2.2.2 to 2.2.4 provide insights about properties of our proposed method. Nevertheless, they were not set up in a purely unsupervised manner, and the parameter k remained unknown. Therefore the results were not directly applicable to a real-world scenario.

In this section we address these limitations and increment the MVG training pipeline with other components to make it more robust. Namely, we introduce:

1. synthetic anomalies in the $\mathcal{I}_{\text{opt}}$ set to guide the feature selection process;
2. a new objective function designed to leverage the information from this scheme;
3. a k-choice estimation using a simple and effective strategy based on our observations;
4. and an image augmentation step to extend the training set.

2.4.1 Synthetic Anomaly Generation

The synthetic anomaly generation process used in this experiment is inspired on the DRÆM method [Zavrtanik, 2021a]. Fig. 4.14a shows the generation pipeline, which consists of blending a normal image with a texture image with a mask that defines the anomalous regions. The notation in this paragraph exceptionally diverges and refers to the notation in Fig. 4.14a. A

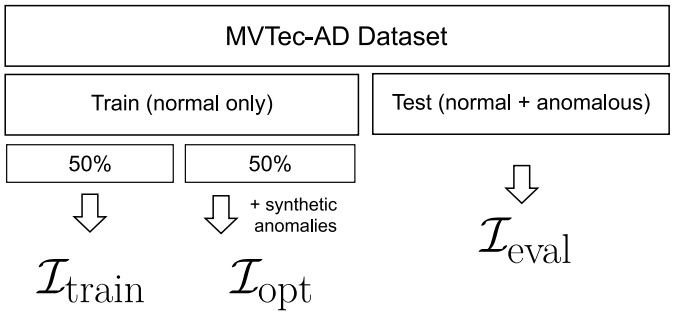


Figure 4.15: Data split for Experiment 4.

random 2D array P (same resolution as the input image) is generated with Perlin noise using random parameters, then binarized with a random threshold to create an anomaly mask M^a. The random parametrization of this process ensures that the shape, size, and density of anomalous regions are varied. The mask M^a is used to blend the original (normal) image $\mathbf{I}^n$ with a randomly selected texture image A such that zero-valued pixels are from $\mathbf{I}^n$ and non-zero pixels are from A. The Describable Textures Dataset (DTD) [Cimpoi, 2014] is used to source the textures, ensuring that the anomalies are sufficiently distinct from the normal images. To further enhance variability, random augmentations, inspired by RandAugment [Cubuk, 2020], are applied to A: three augmentation functions are randomly picked from a set including posterize, sharpness, solarize, equalize, brightness change, color change, and auto-contrast. The two image sources are finally combined:

$$\mathbf{I}^a = \mathrm{M}^a \odot \mathbf{I}^n + (1-\beta)\cdot(\mathrm{M}^a \odot \mathbf{I}^n) + \beta \cdot (\mathrm{M}^a \odot \mathrm{A}) \quad , \tag{4.15}$$

where $\odot$ denotes element-wise multiplication, and β is the opacity parameter that controls the blending (sampled from an uniform distribution in the interval $[0.1, 1.0]$), and $\mathbf{I}^a$ is the synthetic anomalous image. By default, our pipeline generates five anomalies per normal image, and the same anomaly (*i.e.* M^a and A) is never used twice on the same image.

Objective Function (g) We introduce a new objective function for this experiment. Since each anomalous image is generated from a normal image, we leverage this information to build an image-wise signal – as opposed to a set-wise statistic like AUROC. The intuition of the MVG model is that an anomaly detection-effective feature extractor maps anomalous images far from the mean of the normal images. In the whitened feature space, this corresponds to $\mathbf{w}$ vectors with "large" norms (*i.e.* larger than those of the normal images). With a good component selection, one should observe an increase in the norm of $\mathbf{w}$ after the synthetic anomaly is blended with the normal image (Fig. 4.14b). Thus, we use the average difference of Mahalanobis distances between anomalous images and their corresponding normal images as the objective function in the feature selection process:

$$\frac{1}{N_{\text{opt}}} \sum_{i=1}^{N_{\text{opt}}} \| \mathbf{w}_i^a \|_2 - \| \mathbf{w}_i^n \|_2 \quad , \tag{4.16}$$

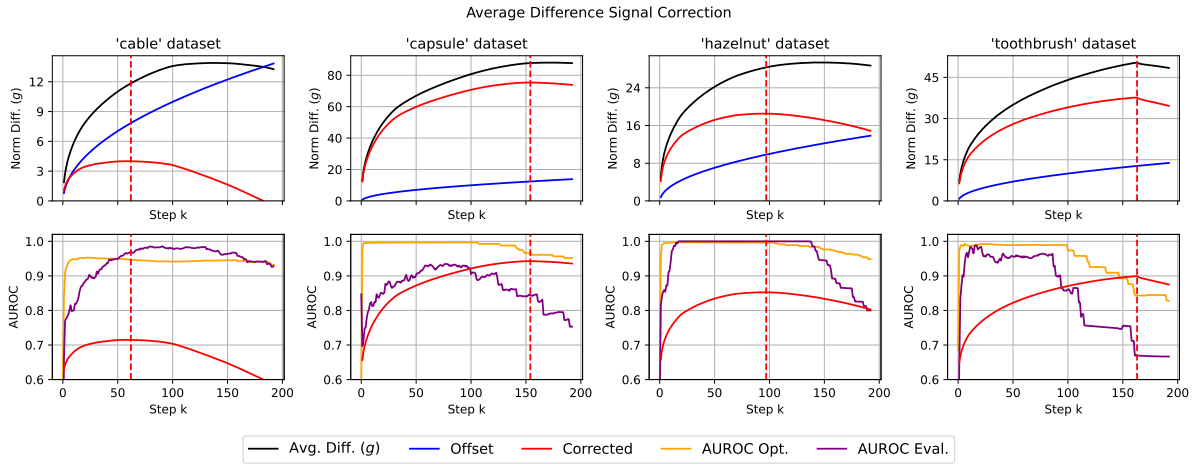


Figure 4.16: (Upper Row) Offset correction of the average difference of Mahalanobis distances between anomalous and normal images. (Lower Row) Corrected objective function *vs.* performance (in terms of AUROC) on the optimization and evaluation sets. Best viewed in color in digital version.

where $\mathbf{w}^a$ and $\mathbf{w}^n$ are, respectively, the whitened feature vectors extracted from images $\mathbf{I}^a$ and $\mathbf{I}^n$ such that $\mathbf{I}^a$ is a synthetic anomalous image generated from a normal image $\mathbf{I}^n$, and N_{opt} is the number of anomalous images in the (eventually augmented) optimization set. Notice that the Mahalanobis distance here was replaced by its equivalent form explained in Section 2.1.1 and Eq. (4.9).

Data Split The data split for this experiment is illustrated in Fig. 4.15. For each MVTec-AD dataset, half of the normal images are used for training the MVG model, and the other half are used for the GreedyES algorithm, and they are respectively denoted $\mathcal{I}_{\text{train}}$ and $\mathcal{I}_{\text{opt}}$. The test set is used for evaluation ($\mathcal{I}_{\text{eval}}$). Note that, unlike the previous experiments, the evaluation set $\mathcal{I}_{\text{eval}}$ is not used in this experiment at all, so the model does not have access to any anomalous images during training pipeline.

2.4.2 Choice of k

Unlike the previous experiments, we take an extra step to select k – thus choosing an actually applicable model. In previous works where (N)PCA was used, the choice of k was based on the ratio of explained variance, and the reference values of the latter were based on any principle or empirical observation. We circumvent this issue and see the choice of k as model parameter that needs to be optimized. The next paragraphs describe two strategies we used to select k.

Maximum Average Distance Difference The first strategy is to select k such that it maximizes the objective function defined in Eq. (4.16) – *i.e.* the peak of the curve k *vs.* $g(k)$. Some examples of this curve are shown in Fig. 4.16 (black curves). This corresponds to selecting the intermediate step in the greedy search that maximizes the average difference signal. However, this strategy is problematic because, as k increases, $\| \mathbf{w} \|_2$ tends to increase (thus, so do the dif-

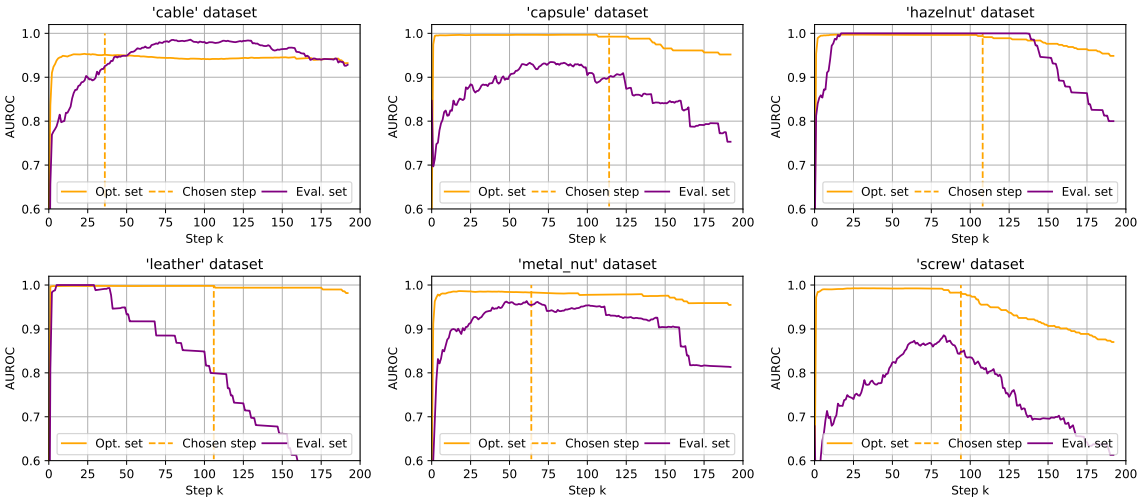


Figure 4.17: Choice of k for Experiment 4 with synthetic anomalies. Best viewed in color in digital version.

ferences in Eq. (4.16)). This happens because the vectors $\mathbf{w}$ are in increasingly high-dimensional spaces, so their norm tend to increase with the dimensionality. To correct this bias, we subtract an offset inspired on a theory-based intuition. Since $\mathbf{w}^n$ vectors are obtained from a whitening operation, their coordinates are uncorrelated and follow a standard normal distribution (although numerical errors cause deviations). Therefore, their norms follow a χ distribution with k degrees of freedom. Thus, the expected value of $\| \mathbf{w}^n \|_2$ is

$$\sqrt{2}\,\frac{\Gamma\left(\frac{1}{2}\,(k+1)\right)}{\Gamma\left(\frac{1}{2}\,k\right)} \quad , \tag{4.17}$$

which, for large k, is approximately $\sqrt{k-\frac{1}{2}} \approx \sqrt{k}$ (blue curves in Fig. 4.16). Based on this fact, we subtract Eq. (4.17) from Eq. (4.16) (*i.e.* black curves minus blue curves) to correct the bias. We refer to the result as the "corrected" average difference signal, represented in red in Fig. 4.16. Finally, we select the k value at the peak of the corrected curve (dotted red vertical lines in Fig. 4.16). Notice, however, that the $\| \mathbf{w}^a \|_2$ term in Eq. (4.16) does *not* necessarily follow a χ distribution, so this correction is rather a heuristic. Despite the correction this strategy is not robust. The outcomes yielded by maximizing the corrected objective function – notice in Fig. 4.16, lower row – are not consistent with the peack of AUROC scores. We omit this strategy in the results section because our second strategy was systematically better.

Empirical Choice of k The second strategy to select k we present is simple yet effective. Given the maximum AUROC score seen in the greedy search AUROC*, we select the highest k such that the AUROC is within a 10% margin of AUROC*:

$$\max \{k \,|\, \text{AUROC}(k) \geq 0.9\,\text{AUROC}^*\} \quad , \tag{4.18}$$

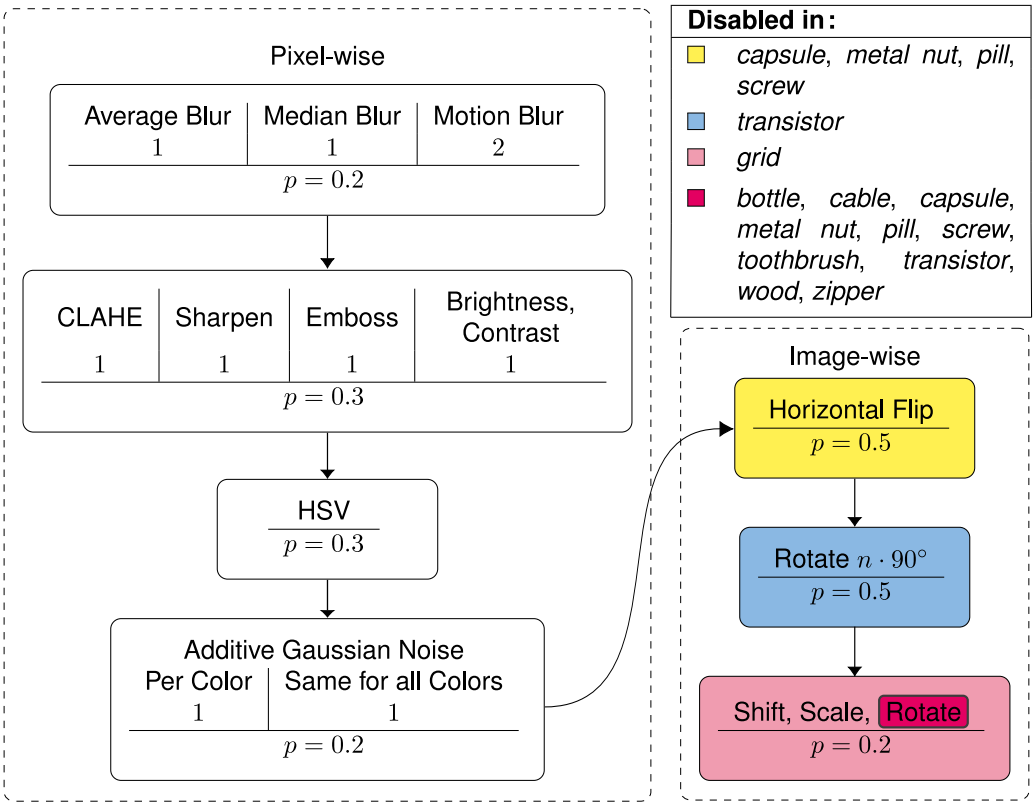


Figure 4.18: Image augmentation pipeline. Diagram extracted from [Rippel, 2021a] (edited).

where AUROC(k) is measured in the optimization set. Intuitively, this choice of k corresponds to the rightmost point of the plateau regime in the k-vs-AUROC curve, but the 10% margin allows for a more robust selection. Some examples of this choice are shown in Fig. 4.17. It generally yields reasonable or close-to-optimal results, with a few problematic cases where the AUROC curves differ too much like on dataset "leather". Notice that the purple curve is the AUROC curve on the *evaluation* set, so that information is not available at training time (nor is it used to select k).

2.4.3 Other Experimental Settings

In the first three experiments (Sections 2.2.2 to 2.2.4), our goal was to understand the behavior of the MVG model and what we could achieve by combining GreedyES on top of it. Unlike those, the focus of this experiment is to achieve the best possible performance on the benchmark datasets. Along with the synthetic anomaly generation and the choice of k, we introduce other elements in the pipeline to improve the performance of the model.

Image Augmentation In addittion to the previously mentioned strategies, we introduce image augmentation to the pipeline to improve the robustness of the model. We used the image augmentation scheme proposed in [Rippel, 2021a] to increase the number of (normal) training samples. The pipeline of augmentations (Fig. 4.18) is conditional on the dataset so that their respective assumptions of normality are preserved. When using augmentations, the training set

of the MVG model and of the feature selection are iterated over n_{augm} epochs. In other words, each original (normal) image generates n_{augm} augmented images. Then, five distinct synthetic anomalies are generated for each of these augmented images. We used $n_{\text{augm}} \in \{0, 1, 3, 10, 30\}$ in our experiments (the case $n_{\text{augm}} = 0$ corresponds to the absence of augmentation). It was observed that beyond 10 the effect of the augmentation becomes generaly negligible. Below this value, the performance has increased variance, making it harder to even take conclusions about the effect of all the other elements in the pipeline. Thus, we simplify the results (Section 2.4.4) by only considering $n_{\text{augm}} \in \{0, 10\}$.

Sequential Forward Floating Selection Since GreedyES makes a per-step optimization (recall the SFS algorithm described in Algorithm 1), it is not guaranteed to find the an optimal solutions. To mitigate this issue, we tested the Sequential *Floating* Forward Selection (SFFS) algorithm [Ferri, 1994b]. This extension allows the optimization to take steps *backwards* in the optimization process (*cf*. numerical example in Fig. 4.19). Our experiments showed that SFFS generally improves the performance of the first 10s of components. As k increases, the improvement becomes less significant, especially in cases with long plateaus. The reason for this is that the in-plateau components are redundant with the most useful ones. Thus, when there are many of these available, the suboptimal choices in SFS do not have a significant impact on the final performance because, at later steps, the algorithm ends up selecting the optimal components. Finally, when the k selection step is applied (*cf*. Section 2.4.2), the improvements of SFFS are compensated, often not significant or even detrimental relative to the solution obtained by GreedyES using vanilla SFS. Based on the results from numerous runs, we concluded that using SFFS is *not* useful in this context, thus the details are deliberately omitted here for brevity.

Feature Extractor We simplify this experiment by only considering the layer `features.6` of EfficientNetB0 as the feature extractor. This layer was selected because Experiment 1 (Section 2.2.2) showed that it should be possible to consistently near-perfect AUROC scores with a small number of eigencomponents in almost all datasets (except "grid"). The deeper nodes also satisfy this condition, but they are in larger feature spaces, which makes the optimization process more computationally expensive.

Baselines For the sake of comparison, we compare GreedyES with two baselines for dimensionality reduction: random selection and NPCA. Random selection consists of sequentially choosing eigencomponents at random – the selection at $k-1$ is contained in the selection at k. It is equivalent to setting the objective function g to a random variable without any conditioning on the data. NPCA is equivalent to setting the objective function g to the *minus* explained variance of the eigencomponents. Both baselines are combined with the same pipeline of image augmentation, synthetic anomaly generation, and k selection.

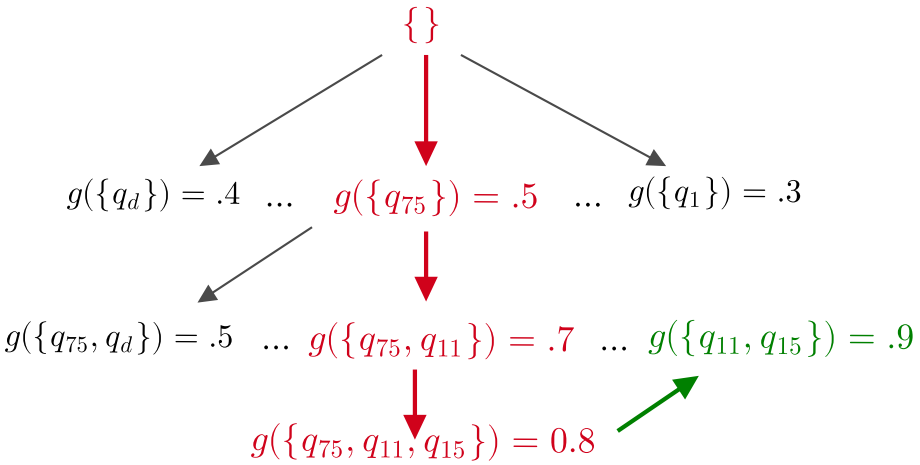


Figure 4.19: Sequential Forward *Floating* Selection (SFFS) [Ferri, 1994b]. Refer to Algorithm 1 and Fig. 4.5 for comparison (forward-only algorithm). SFFS allows previous solutions to be replaced by alternatives with higher performance. Everytime the optimization makes a *forward* step (red; *i.e.* a new component/white feature is added), the algorithm checks if the performance can be improved by removing one of the components – the *backwards* step (green). In the toy example above, the feature selection at $k = 3$ yielded by the forward steps is $\{\mathbf{q}_{75}, \mathbf{q}_{11}, \mathbf{q}_{15}\}$. This solution can be improved by taking a backwards step, removing $\mathbf{q}_{75}$, thus keeping $\{\mathbf{q}_{11}, \mathbf{q}_{15}\}$. This alternative solution at $k = 2$ is better (according to g) than the one obtained by the forward steps ($\{\mathbf{q}_{75}, \mathbf{q}_{11}\}$), so the former replaces the latter. Once the backwards steps cannot improve the solution, the algorithm goes back to the forward steps.

2.4.4 Results

Every experiment was repeated 10 times with different random seeds. Table 4.2 shows the results of all the scenarios in Experiment 4 averaged across seeds with their respective standard error of the mean (SEM) shown in parentheses. Table 4.1 shows a summary of the average AUROC scores across all datasets of the MVTec-AD collection. The rest of this section discusses these results.

Are the Synthetic Anomalies Effective? Fig. 4.20a shows that the AUROC obtained in the optimization set decently correlates with the AUROC in the evaluation set in datasets like "bottle", "cable", and "leather". However, other datasets like "toothbrush" and "capsule" show a significant discrepancy between the two. This suggests that the synthetic anomalies are effective in some cases but not in others. The real anomalies are not defined in advance, so no prior or learned information is known about them. This is a common issue in AD tasks, and it is the reason why it is hard to estimate g even based on a large number of varied synthetic anomalies. Our expectation is that having synthetic anomalies with large enough variance should "dilute" the effect of specific anomalies. While it seems to be a good strategy in some cases, it is not a silver bullet.

Which Dimensionality Reduction Strategy is Better? Experiment 1 showed that GreedyES could theoretically find the optimal subspaces in the feature space with high AD performance.

However, this experiment shows it is very hard to achieve such a result in practice. As a matter of fact, when combined with the empirical choice of k, there is often no significant difference between GreedyES and NPCA as shown in Table 4.2 – sometimes even random selection performs comparably. However, the results of GreedyES are more consistently higher on average than the other strategies. The improvements from the augmentations and from the dimensionality reduction are not mutually exclusive, so they can be combined to achieve better results – in particular, reducing the variance of the AUROC scores thanks to the augmentations. The best results are obtained by combining both, but it is not competitive with the SOTA models. While having a disappointing overall performance, our experiments show that the methodology proposed is a useful tool to improve the MVG model without increasing its computational complexity. Relative to the baseline (no dimensionality reduction), Table 4.1 shows that GreedyES is the best strategy to reduce the dimensionality of the feature space. Fig. 4.20b reveals that, indeed, when attempting to choose eigencomponents using synthetic anomalies, GreedyES has a slight advantage over the other strategies.

What is the Impact of the Image Augmentation? As observed in the previous paragraph, image augmentation often had a similar impact to dimensionality reduction, and sometimes complementary when analizing the average results. However, adding augmentations also helps to decrease the variance. In most of the cases, the cross-seed variance decreased or remained the same when augmentations were added.

Conclusion While the average results (Table 4.1) are not inline with the SOTA, that is it is mostly due to a few datasets like "grid" and "screw" where the performance is substantially lower. In other datasets the results are competitive with SOTA models.

While suboptimal, the MVG model is very simple and the training methodology requires low computational resources compared to other models. Besides, it only works by modelling a probability density function on top a frozen CNN, meaning that it can work along side with the original model, not needing modification.

The results of this experiment show that the GreedyES algorithm is a useful tool to improve the MVG model. The techniques presented here successfully provided means to improve the MVG model without increasing its computational complexity. This makes the model a useful alternative for use cases where more complex techniques cannot be applied or where a fast solution is enough.

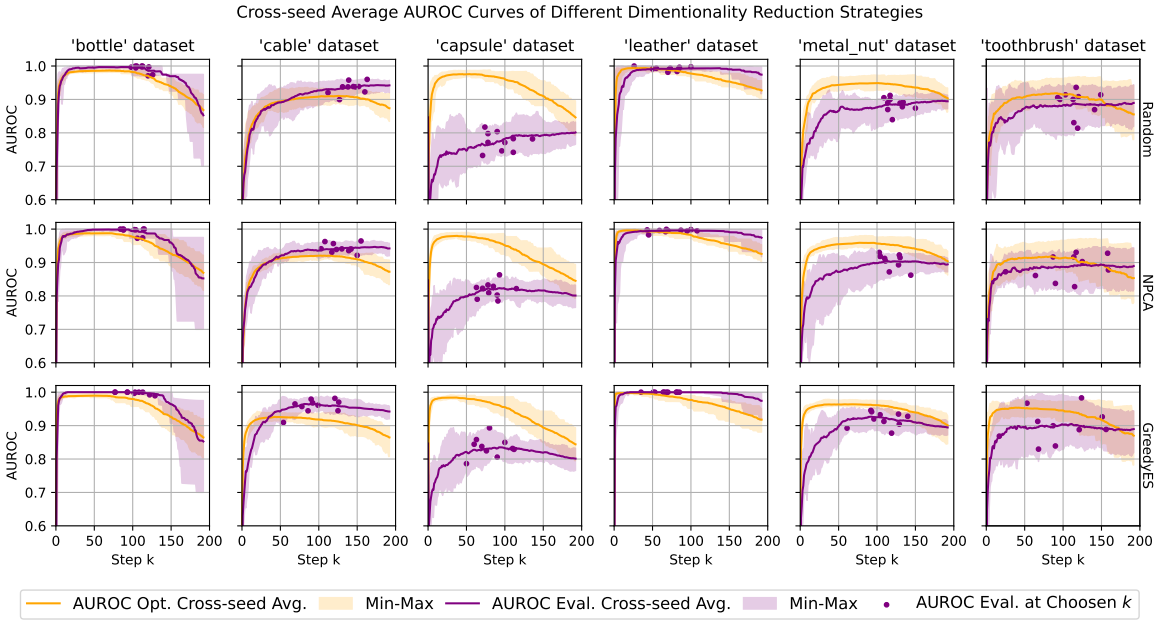
Eigenc. Selec.	$n_{\text{augm}} = 0$	$n_{\text{augm}} = 10$
No Dim. Red.	77.3 (3.0)	90.7 (0.7)
Random	89.0 (2.1)	91.1 (1.2)
NPCA	89.3 (2.4)	91.7 (1.0)
GreedyES	90.7 (2.5)	93.0 (1.0)
GreedyES (B3)	95.5 (2.2)	96.4 (1.1)

Model	AUROC
EfficientAD [Batzner, 2024]	98.8
PatchCore [Roth, 2022]	99.1
RevDist++ [Tien, 2023]	99.4
SimpleNet [Liu, 2023]	99.6

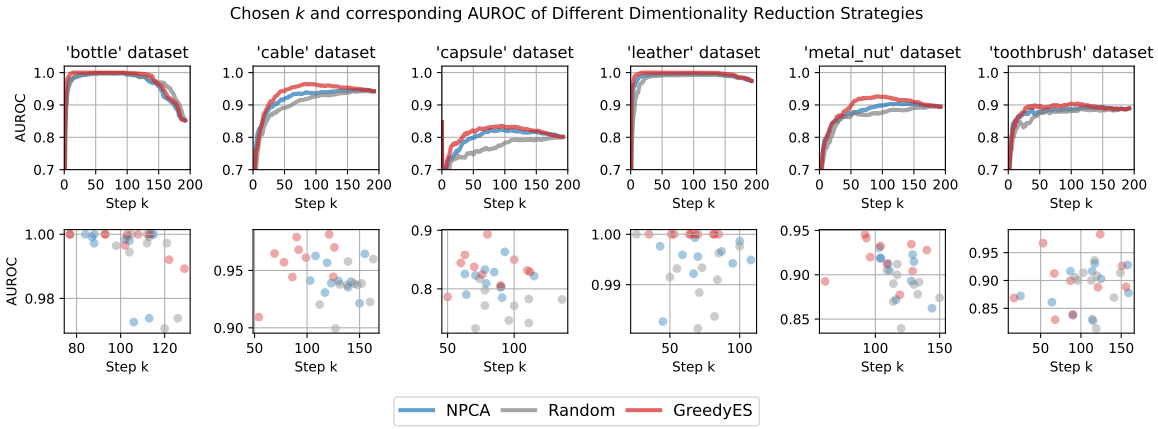
Table 4.1: Results of Experiment 4 with synthetic anomalies, using different dimensionality reduction strategies combined with k selection and image augmentation. Average AUROC scores (%) across all datasets of the MVTec-AD collection. Cross-dataset average of the cross-seed standard deviations in parentheses An additional result using a larger feature extractor (EfficientNetB3) is shown for comparison. Other SOTA models from the benchmark in Chapter 2 are shown for reference on the right.

Dataset	n_{augm}	No Dim. Red.	Random	NPCA	GreedyES
bottle	0	95.2 (0.4)	98.7 (0.5)	98.5 (0.5)	98.9 (0.5)
	10	97.5 (0.00)	99.2 (0.4)	99.4 (0.3)	99.9 (0.07)
cable	0	89.0 (1.1)	95.3 (0.2)	95.6 (0.3)	96.4 (0.7)
	10	96.9 (0.2)	96.2 (0.2)	96.1 (0.2)	98.3 (0.3)
capsule	0	79.6 (1.2)	88.9 (0.7)	89.2 (1.1)	91.0 (0.8)
	10	84.3 (0.2)	84.5 (0.4)	85.5 (0.4)	87.3 (0.5)
carpet	0	67.1 (1.4)	78.1 (0.2)	78.2 (0.07)	79.0 (0.2)
	10	99.4 (0.02)	99.1 (0.1)	99.4 (0.09)	100.0 (0.00)
grid	0	54.9 (0.9)	55.8 (1.0)	61.6 (1.1)	63.8 (1.3)
	10	58.1 (0.3)	59.0 (0.7)	66.0 (0.5)	67.0 (0.6)
hazelnut	0	81.4 (0.8)	99.1 (0.4)	99.4 (0.3)	100.0 (0.05)
	10	99.8 (0.03)	99.7 (0.1)	99.9 (0.02)	100.0 (0.00)
leather	0	53.6 (1.3)	77.8 (1.7)	76.4 (1.9)	76.9 (2.2)
	10	98.0 (0.2)	99.7 (0.04)	100.0 (0.01)	100.0 (0.00)
metal nut	0	81.6 (1.2)	92.6 (0.2)	93.2 (0.4)	96.1 (0.5)
	10	92.4 (0.3)	91.4 (0.4)	93.1 (0.3)	94.5 (0.5)
pill	0	89.1 (0.3)	89.7 (0.4)	90.9 (0.4)	92.1 (0.4)
	10	89.6 (0.2)	88.5 (0.4)	90.8 (0.3)	92.5 (0.4)
screw	0	60.7 (1.7)	83.1 (0.9)	82.1 (1.2)	84.8 (1.3)
	10	77.9 (0.4)	76.5 (0.9)	69.2 (1.1)	74.5 (0.8)
tile	0	88.8 (1.1)	99.9 (0.03)	99.8 (0.2)	100.0 (0.00)
	10	100.0 (0.01)	99.7 (0.1)	99.9 (0.03)	100.0 (0.00)
toothbrush	0	66.6 (1.0)	91.3 (1.2)	88.8 (2.0)	91.3 (1.6)
	10	91.1 (0.6)	89.9 (0.8)	90.3 (0.7)	91.9 (0.6)
transistor	0	86.6 (0.9)	94.5 (0.5)	94.7 (0.3)	96.5 (0.3)
	10	93.1 (0.4)	93.2 (0.5)	94.7 (0.3)	96.2 (0.3)
wood	0	73.6 (0.3)	95.3 (1.6)	96.2 (1.3)	97.5 (1.4)
	10	90.8 (0.4)	99.2 (0.08)	99.5 (0.05)	100.0 (0.00)
zipper	0	92.3 (0.6)	94.5 (0.4)	95.2 (0.5)	96.5 (0.6)
	10	91.2 (0.2)	90.0 (0.4)	91.3 (0.5)	92.3 (0.6)

Table 4.2: Results of Experiment 4 with synthetic anomalies, using different dimensionality reduction strategies combined with k selection and image augmentation. Cross-seed average AUROC scores (%) and their standard error of the mean (SEM) for each dataset of the MVTec-AD collection.



(a) Optimization *vs.* evaluation AUROC.



(b) Choices of k for different dimension reduction strategies.

Figure 4.20: (a) Optimization and evaluation AUROC curves for different dimensionality reduction strategies and across all seds (min, max, and mean). (b) Evaluation AUROC average curves (on the top) and obtained AUROC scores of the chosen k (on the bottom). Best viewed in color in digital version.

3 Visualization

In this section we present two visualization strategies for MVG-based models. Our goal is to provide means to qualitatively analyze the model's predictions. These tools can be used to understand the behavior of the underlying CNN and the features extracted by it. Specifically, one can plug this methodology on top of any CNN – eventually trained for a different task – to locallly understand/debug the features in a classe's neighborhood.

In section Section 3.1, we present our work [Bertoldo, 2023c] on patch-wise models. Recall the patch-wise feature extraction in Fig. 4.1c. The visualizations reveal, for instance, plausible explanations for the results seen in Section 2.3.4. Namely, spurious anomaly scores due to multiple modes feature space distribution, for instance, due to border effects or object-background separation.

In section Section 3.2 we present an explanation method for the image-wise models discussed in Section 2. A decomposition of the Mahalanobis distance score enables us to visualize the contribution of each eigencomponent and each pixel to the final anomaly score. This extends the image-wise model's capabilities to also localize the anomalies in the image.

In both cases, we exploit the equivalence between the Mahalanobis distance score and the norm of the whitened feature vector (Eq. (4.9)) to assign a score to each individual pixel/eigencomponent in the feature map.

3.1 Visualization of Patch-wise Models

Unlike the previous sections, we experiment with patch-wise models: instead of aggregating the feature maps into a single vector, we extract all the feature vectors from it (*cf*. Fig. 4.1c). The vectors $\mathbf{x}$ and their corresponding $\mathbf{w}$ correspond to patches in the image, not to the whole image. Since they correspond to patches in the image, one can generate an anomaly score map from the Mahalanobis distance scores at each spatial index of the feature map. Let $\phi_l(\mathbf{I}) = \mathbf{X} \in \mathbb{R}^{M \times d}$ be the feature map outputed by the CNN for an image $\mathbf{I}$. The spatial resolution M[6] is specific to the layer l and is lower than the resolution of the input image. Given the vector $\mathbf{X}_j \in \mathbb{R}^d$, at pixel position j, the anomaly score of its corresponding patch is:

$$\text{MahaDist}_l\left(\mathbf{X}_j\right) = \| \boldsymbol{\Sigma}^{-1}\left(\mathbf{X}_j - \boldsymbol{\mu}\right) \|_2 = \| \mathbf{W}_j \|_2 \quad , \tag{4.19}$$

where $\mathbf{W}_j = \boldsymbol{\Sigma}^{-1}\left(\mathbf{X}_j - \boldsymbol{\mu}\right)$ is the whitened version of the j-th pixel's $\mathbf{X}_j$. In practice, all $\mathbf{x}_j$ are computed for all j at once to build a map of Mahalanobis distances $\mathbf{s} \in \mathbb{R}_+^{M \times d}$ (*cf*. right most columns in Figs. 4.26 and 4.27), also referred to as *anomaly score map*. This is a simplified version of PaDiM [Defard, 2021], where each pixel j has its own parameters $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$. The anomaly maps's entries can be rewritten as

$$\mathbf{s}_j = \| \mathbf{W}_j \|_2 = \sqrt{\sum_{k=1}^{d} \left(\mathbf{W}_{jk}\right)^2} \quad , \tag{4.20}$$

[6] $M = W \times H$, where W and H are, respectively, the image's width and height.

Table 4.3: Anomaly segmentation performances of ResNet-18 on the category "hazelnut" from the MVTec-AD dataset.

Layer	AUROC (%)	AUPRO (%)
layer1	92.78	82.87
layer2	94.48	81.60
layer3	93.95	72.98
layer4	88.63	57.20

where k is the index of a channel (*i.e.* on the feature dimension)[7]. We define the array $\mathbf{W}^2 \in \mathbb{R}_+^{M \times d}$ such that

$$\left(\mathbf{W}^2\right)_{jk} = \left(\mathbf{W}_{jk}\right)^2 = \left(\mathbf{\Sigma}^{-1}\left(\mathbf{X}_j - \boldsymbol{\mu}\right)\right)_k^2 \tag{4.21}$$

so the anomaly score at pixel j can be written as a sum of its entries along the feature dimension:

$$\mathbf{s}_j = \sqrt{\sum_{k=1}^{d} \left(\mathbf{W}^2\right)_{jk}} \quad . \tag{4.22}$$

Since the square function is monotonically increasing, we can simply analyze $\text{MahaDist}_l\left(\mathbf{X}_j\right)^2$ to get rid of the square root without losing or distorting any information. Finally, $\mathbf{W}^2$ is upsampled to match the resolution of the input image (224×224). Each channel k in $\mathbf{W}^2$ is upsampled independently with bilinear interpolation. For the sake of simplicity, we overcharge the notation and refer to $\mathbf{W}^2$ as the one after the upsampling in the following sections.

By defining $\mathbf{W}^2$, we can visualize a map at a given dimension k, noted as $\left(\mathbf{W}^2\right)_{\cdot k}$. This change of perspective is simple but provides two interesting properties for the visualization. First, the values between pixels in the same $\left(\mathbf{W}^2\right)_{\cdot k}$ are directly comparable because the squared function's distortion is already taken into account. Second, the sum of the all the maps $\left(\mathbf{W}^2\right)_{\cdot k}$ is the anomaly score map.

Dataset and Backbone Like in the previous sections, independent models are trained on different layers of the backbone CNN and each dataset from MVTec-AD. All the data and visualizations (discussed in the following sections) are publicly available on Zenodo [Bertoldo, 2023a]. However, we only present the results for the category "hazelnut" for the sake of space and simplicity because similar observations were seen in other categories. Our goal is to explore a visualization method to analyze the feature maps and how they detect (or not) anomalous inputs, so we prefered to focus on a single dataset instead of showing all of them. We used a ResNet-18 [He, 2015] pre-trained on the ImageNet dataset (`torchvision` version `0.15.2`, weights `ResNet18_Weights.IMAGENET1K_V1`) as the backbone CNN to extract feature maps. Its four major layer blocks[8] were used as feature extractors. Their pixel-wise anomaly segmentation performances in terms of AUROC and AUPRO are shown in Table 4.3 for reference.

[7] Not to be confused with k from Section 2.

[8] "layer1", "layer2", "layer3", and "layer4" in `torchvision`'s terminology.

Implementation Details All visualizations consist of a superposition (blending $\alpha = 0.5$) of two sources: the input image $\mathbf{I}$ and the score maps from individual components of $\mathbf{W}^2$, which we refer to as "heatmaps". The whitened maps $\mathbf{W}^2$ are extracted from all the images in the train and the test sets of the dataset, then normalized to generate heatmaps. The color scale "inferno" is used with the minimum value set to zero and the maximum value set to the 99-th percentile percentile of the individual values in all $\mathbf{W}^2$, denoted $W^2_{\mathrm{P}_{99}}$. This normalization is chosen (instead of the maximum value) to avoid a bad visual result on low to mid-distribution values. Using the maximum would result in extreme values in the right tail of the distribution pushing the color resolution away from the bulk of the data. Two normalization scopes – yielding different visual interpretations – for the heatmaps are proposed:

1. **Per-component**: $W^2_{\mathrm{P}_{99}}$ is independently computed for each component (*i.e.* channel in the whitened space). The statistic accounts for all values in the same (*single*) component from all instances. This strategy is useful to understand the information encoded in each component because it provides better visibility of the nuances across different images. Using the same color scale for all components results in some components being visually dominated by others.

2. **Cross-component**: $W^2_{\mathrm{P}_{99}}$ is computed with all components confounded. The statistic accounts for all values from *all* components from all instances. This strategy is useful to understand how the per-component heatmaps are combined to build the final anomaly score map – the latter is the sum of all the former. With this normalization, any two component maps are visually comparable in magnitude.

Results We visually inspected a large number of images and their heatmaps. The visualizations reveal that some weaknesses of the model can be easily identified, but understanding how components – jointly – build a correct anomaly score map is not straightforward. We present a selection of visualizations that summarize the insights learned. The channels of $\mathbf{W}^2$ are re-sorted by their individual AUROC (sorting is done for each layer independently). In other words, each component's heatmap is interpreted as if it were the anomaly score map itself. We present the results in the following sections by splitting the components into two groups: low AUROC and high AUROC. In Section 3.1.1 we discuss the *bad* components (*i.e.* low AUROC), whose weaknesses can be seen even on the train set. In Section 3.1.2 we focus on *good* components (*i.e.* high AUROC) and how they combine to build successful anomaly score maps. Both sections are organized by "topics": resolution mismatch, whitened component value ranges, border effects, and artifacts in the feature maps. These correspond to the most common observed phenomena in the visualizations (including other datasets not shown here).

Layout in Figs. 4.22 to 4.27 The upper title of each figure shows the segmentation performance of the model (*i.e.* layer in the ResNet backbone) being visualized (using all the components). On top of each visualization, the index of the image (in the training or test sets) and the index of the component in $\mathbf{W}^2$ are indicated. Both are 0-starting indexes. The components

indexes refer to the original order of the components in $\mathbf{W}^2$ after the whitening operation (before the sorting by AUROC), so they are sorted from lowest (zero) to highest $(d-1)$ explained variance (*i.e.* singular values).

3.1.1 Low-AUROC components

In this subsection, all heatmaps use the per-component color scale to highlight visual artifacts. Figs. 4.22 and 4.23 display heatmaps of three components among the *worst* (lowest AUROC) from all the four layers of the backbone ResNet. They all show the same three *normal images from the training set*. Figs. 4.24 and 4.25 display the same layers from Fig. 4.22 on *anomalous images from the test set*. Anomalous images from layer3 and layer4 were omitted for brevity; similar observations were made in those layers.

Resolution Mismatch [Defard, 2021] observed that deeper layers from a same backbone (they also used a ResNet) achieve lower performances despite their higher-dimensional feature vectors. A similar result was observed in image-wise models (related to previous sections) in [Lin, 2022; Kim, 2021; Rippel, 2021c; Rippel, 2021a]. These authors hypothesized that this happens because those layers are too biased by the classes from the pre-training task. Our results show an alternative (or complementary) and simpler explanation: the resolution in deeper layers is too coarse to localize anomalies. As downsampling operations are chained in the backbone CNN, the feature maps' resolutions become too coarse. In layer4 (Fig. 4.23b) for instance, the resolution M is 7×7 while the input image is 224×224, thus a pixel in the former represents a patch of 32×32 from the latter.

Range of Values Given the working hypothesis – feature vectors follow a Gaussian distribution – the values in each axis of $\mathbf{w}$ are expected to follow a standard normal distribution. As a consequence, the values in $\mathbf{W}^2$ are expected to be follow a χ^2 distribution with one degree of freedom in the normal images (Fig. 4.22), hence $\approx 99\%$ of the values should be in the interval $[0, 6.7]$. Since we set the upper bound of the color scale from the empirical P_{99}, we would expect it be near 6.7. In Figs. 4.22 and 4.23, however, an inconsistent behavior is observed. In some cases, this seems to be explained by some components showing multiple modes – therefore contradicting the Gaussian assumption – where one mode concentrates far from the mean (*i.e.* high values). Examples: layer1 component 62 ($W^2_{P_{99}} \approx 16$); layer4 component 17 ($W^2_{P_{99}} \approx 16$). Other cases show – again, contradicting the Gaussian assumption – a concentration more compact than expected. Examples: layer1 component 63 and layer2 component 127 (Fig. 4.22) have saturation values $W^2_{P_{99}}$ around 3 to 3.5, which is lower than expected.

Border Effects Several components show artifacts on the borders and corners of the heatmaps. These artifacts have very high activation values and overshadow the rest of the activations. Moreover, they often are input-independent – *i.e.* they happen no matter the content of the input image. Examples of borders artifacts (vertical/horizontal strips): layer1 components 55 and 62;

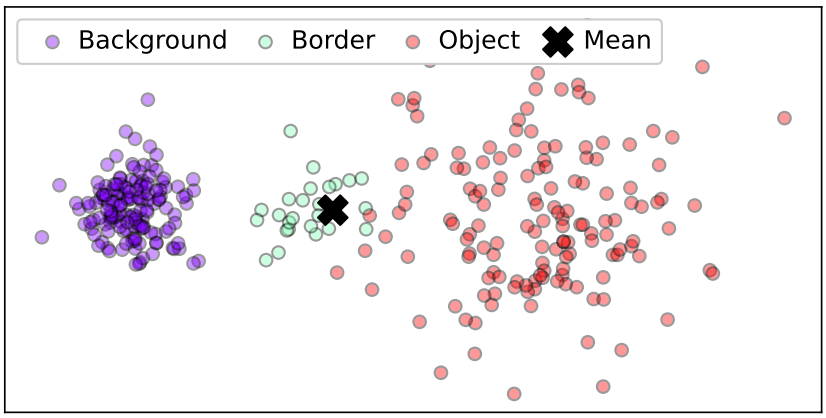
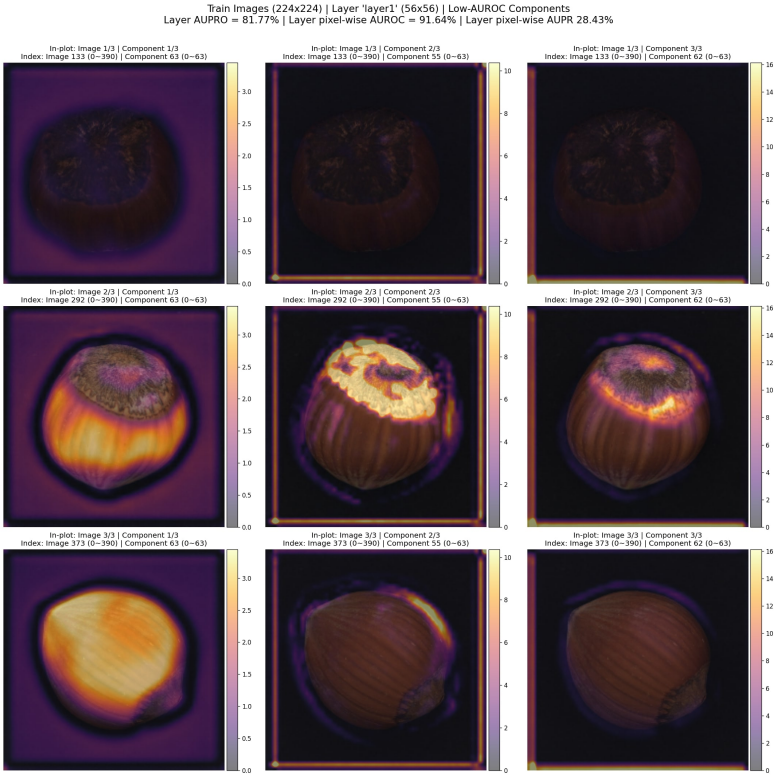


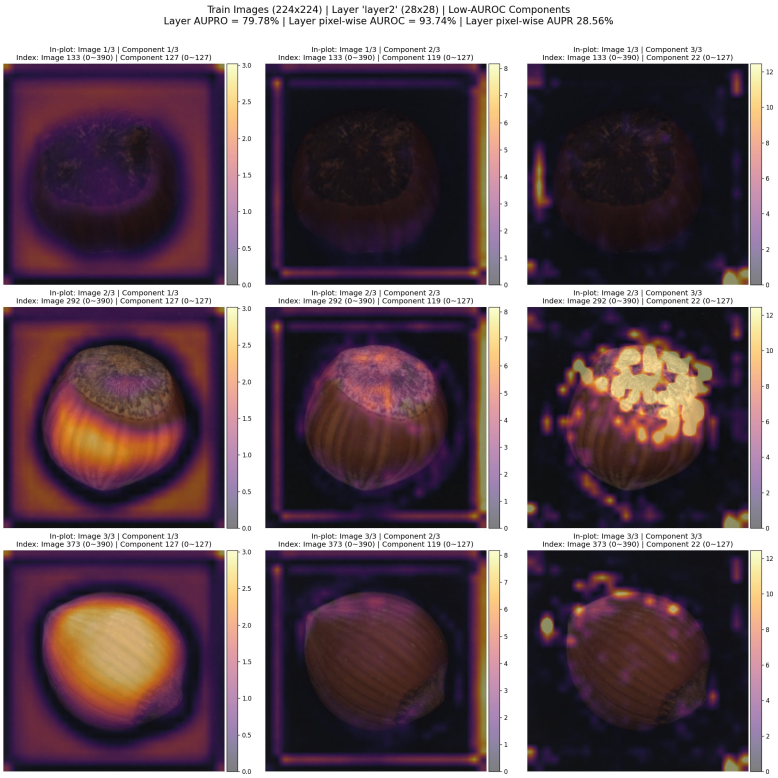
Figure 4.21: Illustration of background, object, and border regions in the feature space. Activation channels with this behavior show as 'contour detectors' (*cf*. Fig. 4.22, left-most heatmaps). The mean of the two modes (background and object) falls in the middle of the feature space, so the vectors from the countours of the object are close to the mean, showing low anomaly scores, while the two modes have high scores.

layer2 component 119; layer3 component 133; layer4 components 510 and 133. Examples of corner artifacts: layer2 component 22; layer3 component 213; layer4 component 17. We hypothesize these issues ensue from the padding used in the backbone CNN. As multiple padded convolutions are successively applied, the border effects accumulate and the feature vectors near the borders shift to a different distribution mode. The visualizations of anomalous images (from the test set) in Figs. 4.24 and 4.25 show that these border effects do not necessarily spoil component entirely. In some cases these spurious activations are overshadowed by higher activations on the actual anomalies. In layer2 component 22, for instance, the anomalies activate more intensely than the corner artifacts, making them practically disappear. Unfornately, more often than not these artifactful activations dominate those anomaly score maps, making them less useful for the AD task. Examples: layer1 component 33 and layer2 component 119; even real anomalies have lower values the border effects.

Object-Background Modes Some components show a behavior as illustrated in Fig. 4.21, particularly in object datasets (like the "hazelnut"). They seem to detect the border between the object (foreground) and the background. Pixels near the object's contour have low values, while the background and the object have high values, because the mean of the two modes falls in the middle of the feature space. Notice that this phenomenon persists in anomalous images (Fig. 4.24), and some anomalies even have lower values than the normal foreground and background (*e.g.* images 46 and 66 in Fig. 4.24), which contradicts the expected behavior of the model. It is worth noting that the border effects and the object-background modes are more pronounced in our model than in PaDiM [Defard, 2021] because the latter has pixel position-dependent parameters $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$. However, this object-background limitation would still occur with PaDiM if the images in the training set are not aligned – some images would have background pixels in the object region and vice-versa.

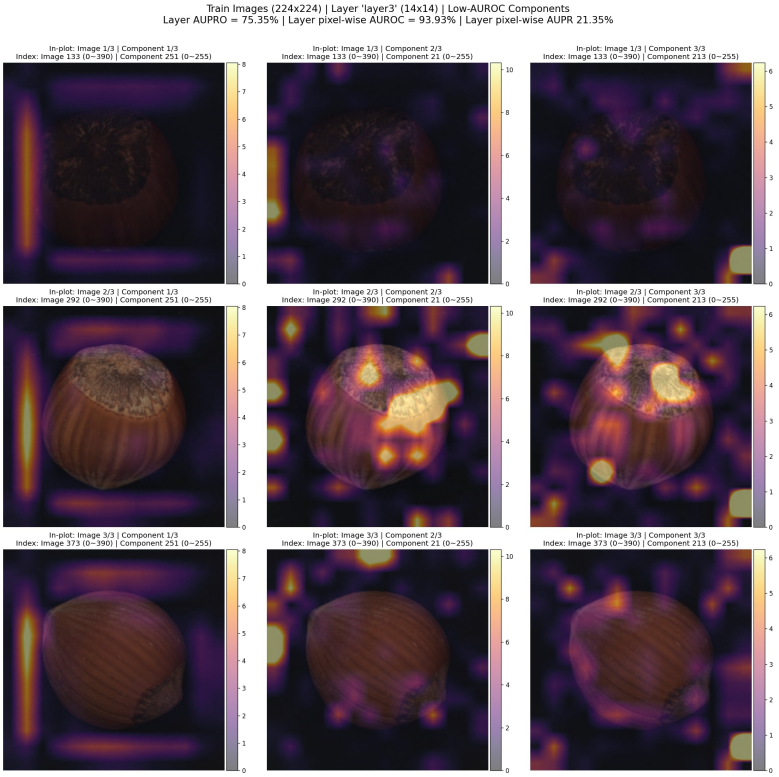


(a) layer1

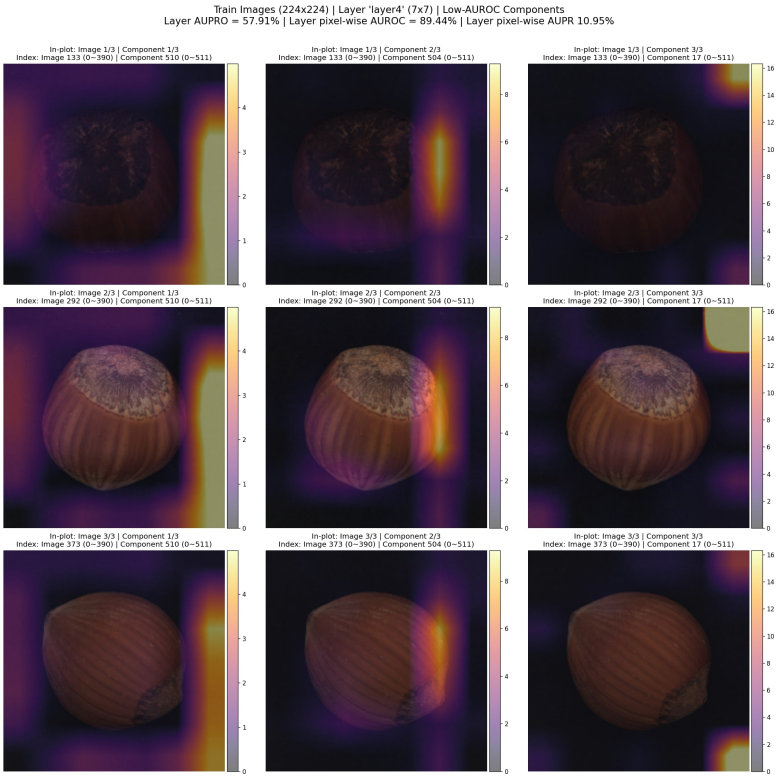


(b) layer2

Figure 4.22: Visualizations on **training (normal)** images (per-component normalization). Selection of **low-AUROC** components.



(a) layer3



(b) layer4

Figure 4.23: Visualizations on **training (normal)** images (per-component normalization). Selection of **low-AUROC** components.

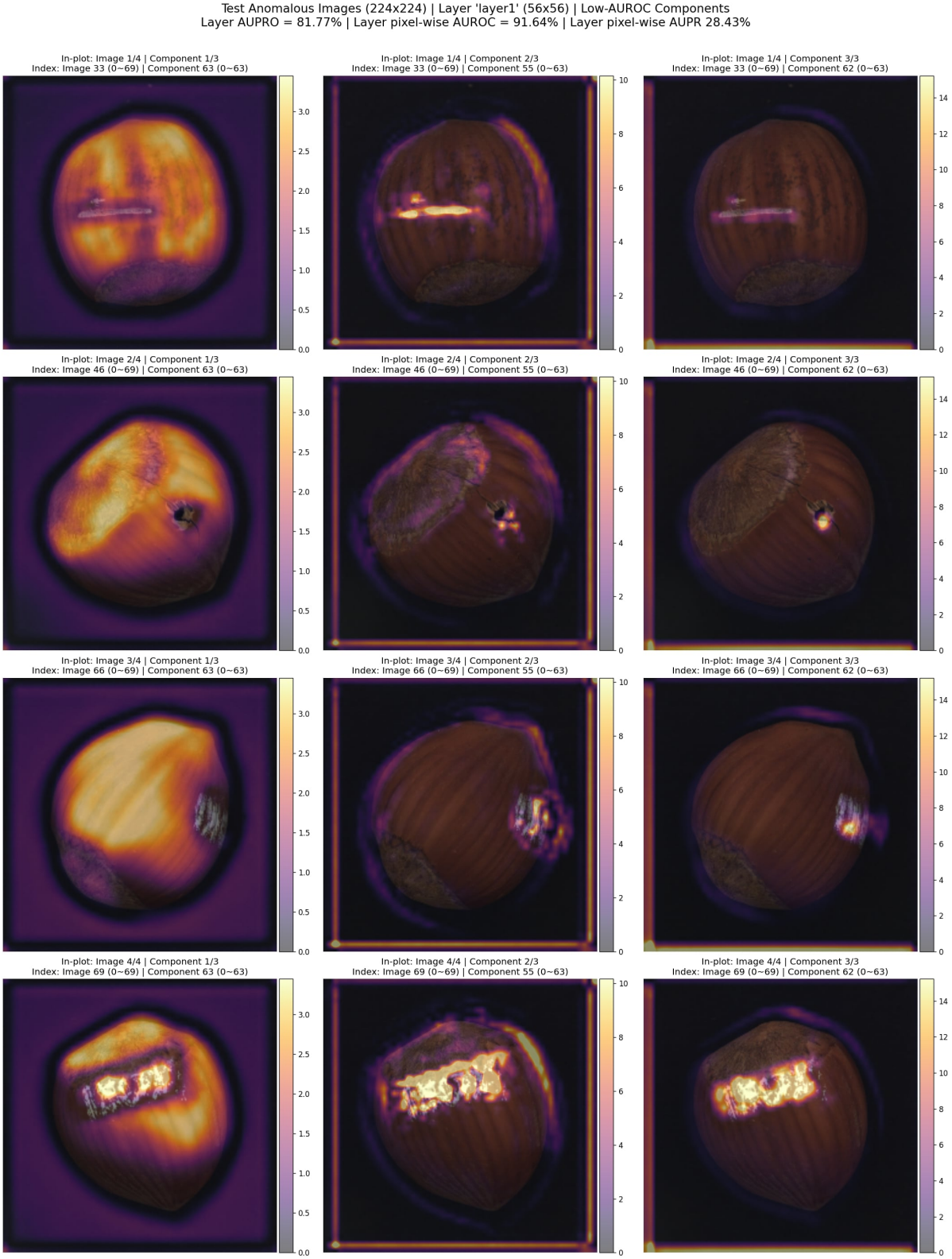


Figure 4.24: Visualizations on test anomalous images (per-component normalization). Selection of **low-AUROC** components on **layer1**.

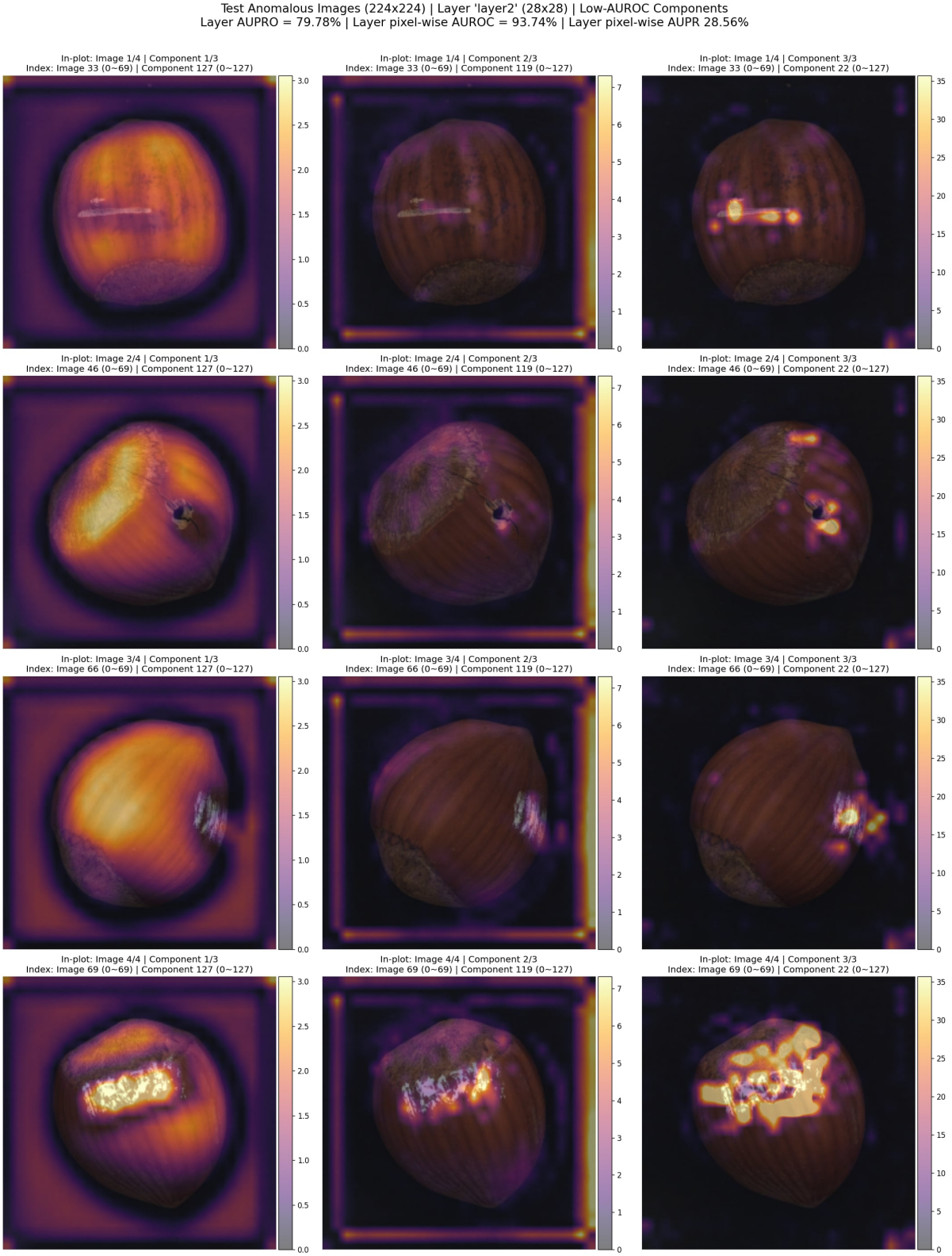


Figure 4.25: Visualizations on test anomalous images (per-component normalization). Selection of **low-AUROC** components on **layer2**.

3.1.2 High-AUROC components

In this section, all heatmaps use the cross-component color scale to better visually compare how the values of different components compare and interact. Figs. 4.26 and 4.27 displays heatmaps of three components among the *best* (highest single-component AUROC) and the anomaly score map (right-most heatmap at each row) from layer1 and layer2. They both show the same four *anomalous images from the test set*. The visualizations from layer3 and layer4 are omitted due to the lack of space but similar phenomena have been observed.

Border/Corner Artifacts The artifacts mentioned in Section 3.1.1 are also present in some cases such as in layer1 components 13 and 14 and layer2 components 74 and 75. However, they all *also* show high values on the *actual* anomalous pixels, which explains why these layers yield better detection performances.

Agreement on Anomalies It can be observed that there is some degree of "agreement" on the anomalous pixels, but rather "disagreement" on the false positive pixels. The high-value regions on the anomalies (thus true positives) intersect across components, redundantly detecting anomalous patches. However, the high-value regions on normal pixels (thus false positives) are randomly placed. The final anomaly score map (sum of these components) confirms this explanation. Since the false positive regions are randomly placed, such regions only have the score accumulated over very few channels. The sum of true positive values, on the other hand, accumulates over multiple channels so it ends up shadowing the total score from normal patches. Future work could investigate how to exploit this redundancy to unsupervisedly select the best components for each image.

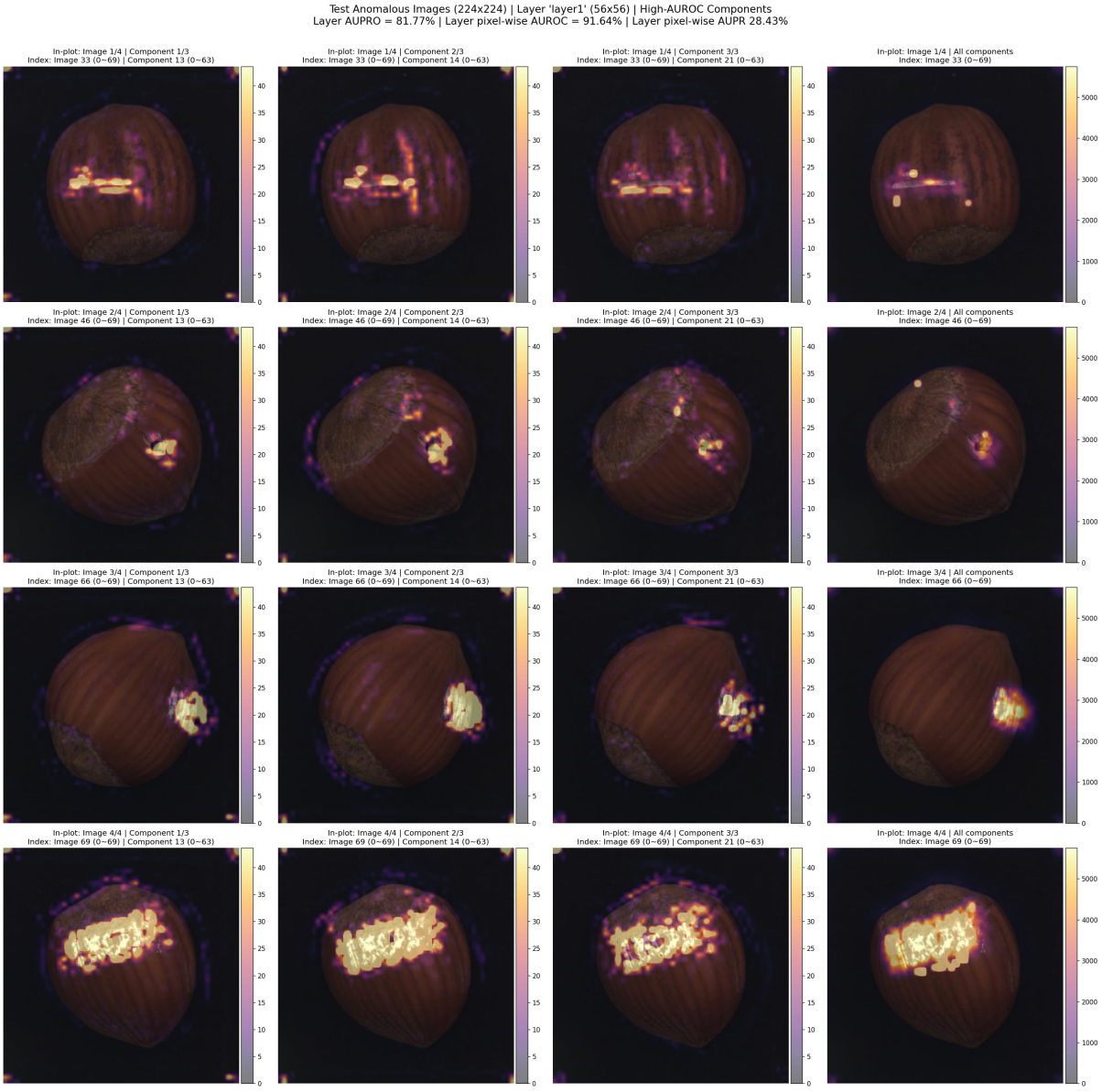


Figure 4.26: Visualizations on test anomalous images (cross-component normalization). Selection of **high-AUROC** components on **layer1**.

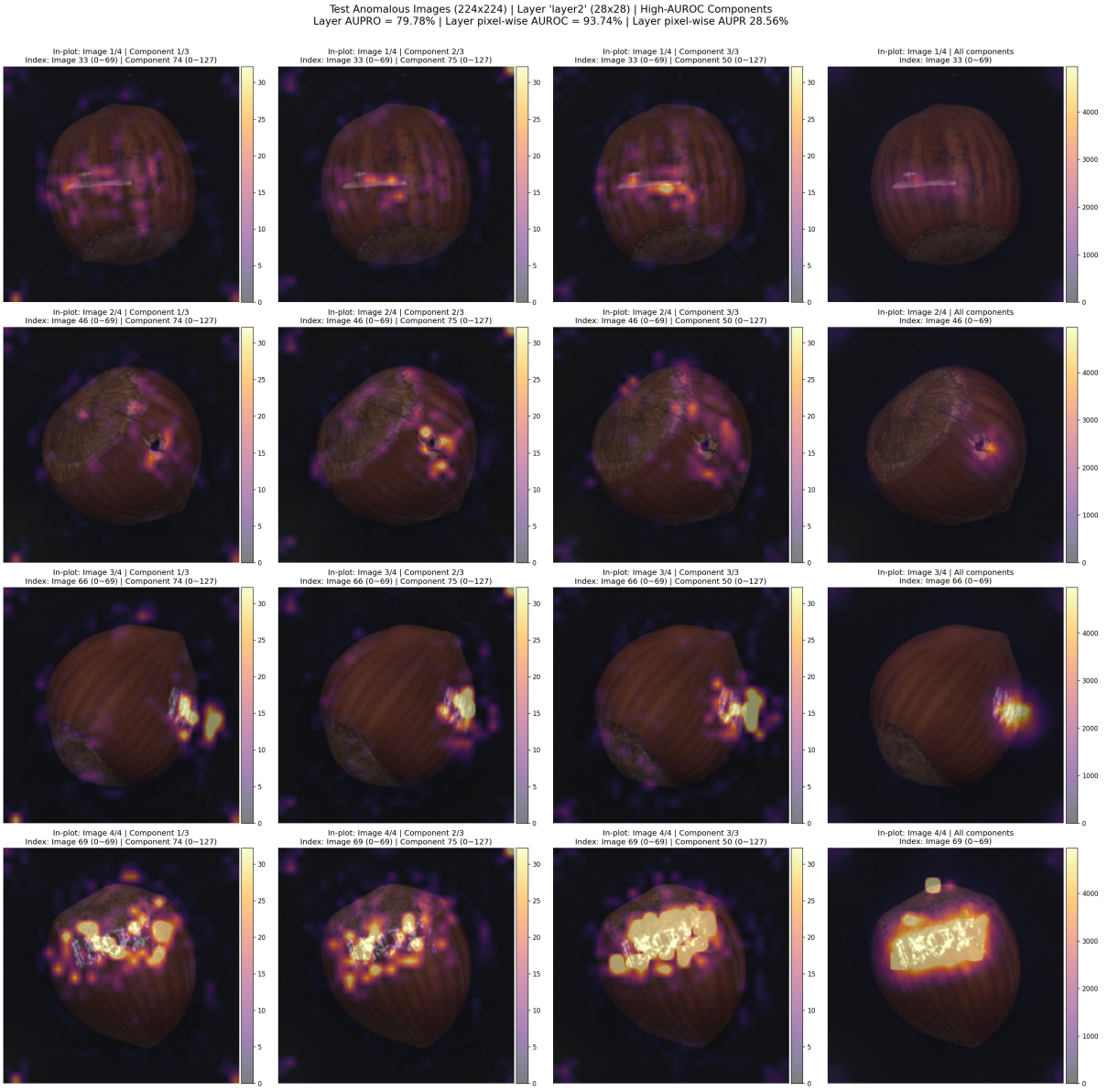


Figure 4.27: Visualizations on test anomalous images (cross-component normalization). Selection of **high-AUROC** components on **layer2**.

3.2 Visualization of Image-wise Models

In this section we introduce a methodology to explain the output score of the image-wise MVG model (Fig. 4.1b), aiming to contribute to its interpretability. This contribution provides means to qualitatively analyze a trained model's inference results – *i.e.* it can be used as a validation tool. Such models (and their variants) based on CNN feature extractors have been used in many previous works, but – to the best of our knowledge – no model-specific explainability method has been proposed.

Our strategy consists of visualizing the contribution of each pixel to the total anomaly score of an image. The rationale of our methodology is to decompose the Mahalanobis distance score into proportional scores at each pixel position j such that their sum is the total score. As detailed later on, a heatmap $\mathbf{C}$ is defined such that the value $\mathbf{C}j$ can be seen as "the percentage of the total score coming from pixel j". The process of assigning contribution scores relies on linear algebra manipulations, so the interpretability of the results is straightforward and does not introduce estimation errors.

All the heatmaps in this section are generated at the feature map's resolution, which is lower than the input image's resolution. When comparing the heatmaps to the input image or to the ground truth annotation, they are upsampled to match the input image's resolution. However, unlike Section 3.1, *nearest neighbor* interpolation is used instead of bilinear interpolation to avoid smoothing the heatmaps. In this section we use, as feature extractor, the outputs of the block "features.6" from EfficientNetB0 [Tan, 2019] pre-trained on the ImageNet dataset.

Whitening and Average Pooling are Commutative First, we show that the whitening operation (recall Section 2.1.1) and the average pooling operator in the image feature extraction are commutative. This enables us to define an activation map $\mathbf{W} \in \mathbb{R}^{M \times d}$ – analogous but not equivalent to the one in the previous section – from where one can access pixel-wise vectors in the whitened space.

Recall that, given an image $\mathbf{I}$ and a feature extractor ϕ_l based on a backbone CNN, the MVG model assigns an anomaly score $s = \text{MahaDist}_l(\phi_l(\mathbf{I})) \in \mathbb{R}_+$ to the image, where MahaDist_l is the Mahalanobis distance using the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ from a set of normal images (recall Section 1.1). The output of ϕ_l is a feature map $\mathbf{X} \in \mathbb{R}^{M \times d}$ (a tensor), where M is its spatial resolution (number of pixels) and d is the number of feature dimensions/axes. Recall that we simplify the two spatial dimensions into a single one (width and height confounded). This feature map is then aggregated with a [global] 2D average pool operation $\text{Pool}_{\text{avg}} : \mathbb{R}^{M \times d} \mapsto \mathbb{R}^d$ to obtain a feature vector $\mathbf{x} \in \mathbb{R}^d$ such that

$$\mathbf{x}_k \;=\; \left(\text{Pool}_{\text{avg}}\left(\mathbf{X}\right)\right)_k \;=\; \frac{1}{M}\sum_{j=1}^{M} \mathbf{X}_{jk} \quad , \tag{4.23}$$

which is then whitened to obtain

$$\mathbf{w} \;=\; \boldsymbol{\Sigma}^{-\frac{1}{2}}\left(\mathbf{x} - \boldsymbol{\mu}\right) \in \mathbb{R}^d \quad , \tag{4.24}$$

the so called "*whitened* feature vector". Recall from Eq. (4.9) that $\text{MahaDist}_l(\mathbf{x}) = \|\mathbf{w}\|_2$.

The average pooling operation was omitted in the notation in the previous sections for simplicity, but we make it explicit here for clarity. Thus the anomaly score s assigned to $\mathbf{I}$ is

$$s = \text{MahaDist}_l\left(\text{Pool}_{\text{avg}}(\mathbf{X})\right) = \|\,\text{Whiten}\left(\text{Pool}_{\text{avg}}(\mathbf{X})\right)\|_2 \quad , \tag{4.25}$$

where $\mathbf{X} = \phi_l(\mathbf{I})$. To create a map of scores, we invert the order of these two operations to obtain a tensor $\mathbf{W} \in \mathbb{R}^{M \times d}$ such that $\text{Pool}_{\text{avg}}(\mathbf{W}) = \mathbf{w}$, as shown below.

From Eq. (4.7), we know that $\mathbf{w} = \text{Whiten}(\mathbf{x}) = \boldsymbol{\Sigma}^{-\frac{1}{2}}(\mathbf{x} - \boldsymbol{\mu})$. It follows that

$$s = \|\,\text{Whiten}\left(\text{Pool}_{\text{avg}}(\mathbf{X})\right)\|_2 = \|\,\boldsymbol{\Sigma}^{-\frac{1}{2}}\left[\text{Pool}_{\text{avg}}(\mathbf{X}) - \boldsymbol{\mu}\right]\|_2 \tag{4.26}$$

$$= \|\,\boldsymbol{\Sigma}^{-\frac{1}{2}}\left[\left(\frac{1}{M}\sum_{j=1}^{M}\mathbf{X}_{j\cdot}\right) - \boldsymbol{\mu}\right]\|_2 \tag{4.27}$$

$$= \|\,\boldsymbol{\Sigma}^{-\frac{1}{2}}\left[\frac{1}{M}\sum_{j=1}^{M}\left(\mathbf{X}_{j\cdot} - \boldsymbol{\mu}\right)\right]\|_2 \tag{4.28}$$

$$= \|\,\frac{1}{M}\sum_{j=1}^{M}\left[\boldsymbol{\Sigma}^{-\frac{1}{2}}\left(\mathbf{X}_{j\cdot} - \boldsymbol{\mu}\right)\right]\|_2 \tag{4.29}$$

$$= \|\,\frac{1}{M}\sum_{j=1}^{M}\mathbf{W}_{j\cdot}\,\|_2 = \|\,\text{Pool}_{\text{avg}}(\mathbf{W})\,\|_2 = \|\,\mathbf{w}\,\|_2 \quad , \tag{4.30}$$

where $\mathbf{W} \in \mathbb{R}^{M \times d}$ is the *whitened feature map*, defined such that

$$\mathbf{W}_{j\cdot} = \text{Whiten}(\mathbf{X}_{j\cdot}) = \boldsymbol{\Sigma}^{-\frac{1}{2}}(\mathbf{X}_{j\cdot} - \boldsymbol{\mu}) \quad . \tag{4.31}$$

The whitened map $\mathbf{W}$ here should not be confused with the one from Section 3.1, where the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ were computed from the patch-wise feature vectors $\mathbf{X}_{j\cdot}$. Here, we refer to the parameters computed from the image-wise feature vectors $\mathbf{x} = \text{Pool}_{\text{avg}}(\mathbf{X})$, as introduced in Section 1.1.

Mahalanobis Distance Map s A straightforward way to visualize the feature's behavior in the whitened map is to compute $\|\cdot\|_2$ at each position j of $\mathbf{W}$ – similar to what was proposed in Section 3.1. Let the map $\mathbf{s} \in \mathbb{R}_+^M$ be defined such that

$$\mathbf{s}_j = \|\,\mathbf{W}_{j\cdot}\,\|_2 = \sqrt{\sum_{k=1}^{d}\left(\mathbf{W}_{jk}\right)^2} \quad , \tag{4.32}$$

so $\mathbf{s}_j$ is the Mahalanobis distance of the patch-wise feature vector $\mathbf{X}_{j\cdot}$. Examples of this map are shown in Fig. 4.32. While this approach is simple to define, it does not directly relate to the anomaly score s because an aggregation function such that $\text{Agg} : \mathbf{s} \mapsto s$ would not be

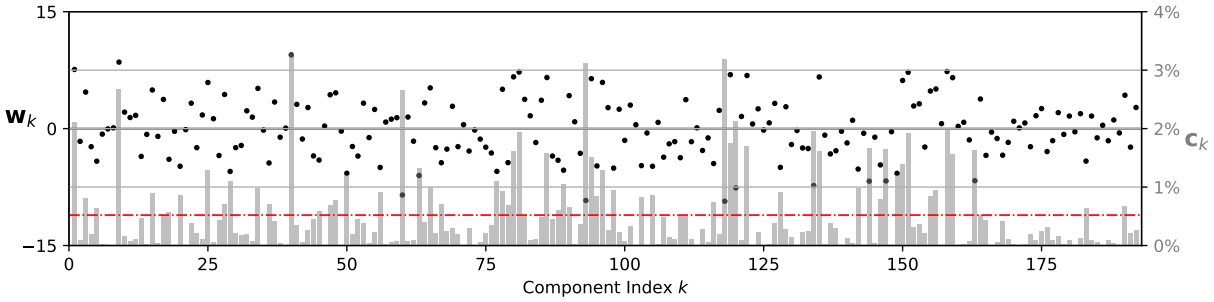


Figure 4.28: Example of the contribution vector $\mathbf{c}$ computed from the whitened vector $\mathbf{w}$. The values in $\mathbf{c}$ are in the interval $[0, 1]$ and sum to one.

straightforward. To address this issue, we define an alternative solution such that the total score s is the sum of the values assigned to each pixel. To achieve this, we introduce the notions of contribution vector $\mathbf{c}$ aforementioned compontent-wise contribution map $\mathbf{c}^{(k)}$, which are used to build a contribution tensor $\mathbf{C}$ that is finally averaged over the channels to obtain a map $\overline{\mathbf{C}}$.

Contribution Vector c Let $\mathbf{c} \in \mathbb{R}^d$, the *contribution vector*, be defined such that

$$\mathbf{c}_k = \frac{\mathbf{w}_k^2}{\sum_{k=1}^{d} \mathbf{w}_k^2} \quad . \tag{4.33}$$

From this definition, it follows that $\mathbf{c}_k \in [0, 1]$ and $\sum_{k=1}^{d} \mathbf{c}_k = 1$. By defining a vector $s\mathbf{c}$, the sum of its entries equals s. Thus a value $\mathbf{c}_k$ can indeed be interpreted as the "percentage of s coming from $\mathbf{w}_k$". Fig. 4.28 shows an example of $\mathbf{w}$ and $\mathbf{c}$.

Component-wise Contribution Map $\mathbf{c}^{(k)}$ Let $\mathbf{c}^{(k)} \in \mathbb{R}^M$, the *component-wise contribution map* of a component k, be – similarly – defined as

$$\mathbf{c}_j^{(k)} = \frac{\mathbf{W}_{jk}}{\sum_{j=1}^{M} \mathbf{W}_{jk}} \quad . \tag{4.34}$$

$\mathbf{c}^{(k)}$ also has the property $\sum_{j=1}^{M} \mathbf{c}_j^{(k)} = 1$. However $\mathbf{c}_j^{(k)} \notin [0, 1]$ because the values in $\mathbf{W}_{\cdot k}$ can have different signs, so the values $\mathbf{c}_j^{(k)}$ cannot be interpreted as a percentage.

This issue is addressed by modifying the entries in $\mathbf{W}_{\cdot k}$ such that only the values "dominating the final score" are kept while the others are replaced by zero. We define a zeroing process such that the target entries cancel out each other in the average pooling operation, so the final score is not altered. In other words, the entries $\mathbf{W}_{jk}$ with highest absolute value and sign equals to $\operatorname{sign}\left(\sum_{j=1}^{M} \mathbf{W}_{jk}\right) = \operatorname{sign}(\mathbf{w}_k)$ are isolated from the rest.

Zeroing Process ($\mathbf{W}_{\cdot k} \mapsto \mathbf{W}^0_{\cdot k}$) By isolating the most relevant entries in $\mathbf{W}_{\cdot k}$, we obtain a modified version $\mathbf{W}^0_{\cdot k}$ where all the non-zero entries have the same sign. This process is illustrated and explained in Fig. 4.29. In summary, the zero-crossing point on the cumulative sum of the sorted values in a $\mathbf{W}_{\cdot k}$ is chosen as *cutoff point*. With the values sorted, only those

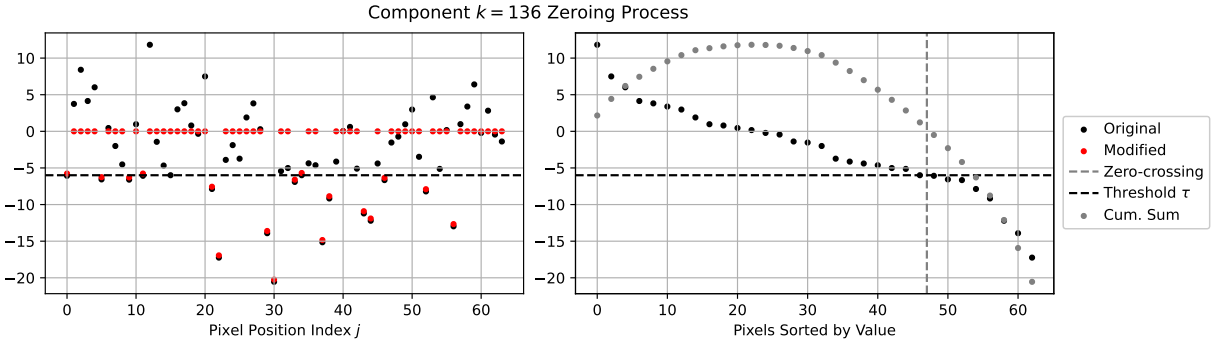


Figure 4.29: Zeroing Process. The following procedure is applied to each map $\mathbf{W}_{\cdot k}$ independently. The plots illustrate the process for a single, fixed component k. On the left, the values j *vs.* $\mathbf{W}_{jk}$ are shown in black. On the right, these values are sorted by value (black), and their cumulative sum – from left to right – is shown in gray (secondary y-axis, scale omitted). From this curve (gray), the zero-crossing point is found (around 47 in the x-axis). The corresponding value on the sorted values (black) defines the threshold τ. Finally, the values coming *before* (*i.e.* above) the threshold are set to zero (left, red). Note that the order is ascending or descending if the sum of the values is, respectively, positive or negative. Correspondingly, the condition to set values to zero is $< \tau$ or $> \tau$ if the sum is, respectively, positive or negative. In the example above, the sum is negative, so the order is descending and the values zeroed out are those $> \tau$. The modified values are shown in red on the left. Notice that a correction (*cf.* Eq. (4.35)) is applied to the non-zeroed values to keep the sum of the map constant, so they are not exactly the same as the original values.

after the cutoff are kept and the other pixels are zeroed. We note $\mathbf{W}^0_{\cdot k}$ and $\mathbf{W}^0$ the zeroed versions of $\mathbf{W}_{\cdot k}$ and $\mathbf{W}$ respectively.

Since the exact cutoff point may fall between two values in the sorted sequence, the average value of a map $\mathbf{W}^0_{\cdot k}$ is slightly different than in $\mathbf{W}_{\cdot k}$. To compensate this, the values in $\mathbf{W}^0_{\cdot k}$ are multiplied by a correction factor

$$\frac{\sum_{j=1}^{M} \mathbf{W}_{jk}}{\sum_{j=1}^{M} \mathbf{W}^0_{jk}} \quad , \tag{4.35}$$

to ensure that $\text{Pool}_{\text{avg}}(\mathbf{W}^0) = \text{Pool}_{\text{avg}}(\mathbf{W}) = \mathbf{w}$. Notice how the non-zeroed values in Fig. 4.29 (on the left) are not exactly the same as the original values, but the relative absolute errors are generally less than 0.2% (*cf.* Fig. 4.30).

Zeroing Effects Fig. 4.31 shows that this modification has the visual effect of "cleaning" the irrelevant part of the signal in each component. By preserving the entries that contribute most to the anomaly score, this increases the contrast of the relevant parts of the signal. When computing the component-wise contribution map $\mathbf{c}^{(k)}$ (*cf.* Eq. (4.34)) from $\mathbf{W}^0$, its values remain in the interval $[0, 1]$. Thus $\mathbf{c}^{(k)}_j$ can be interpreted as the "percentage of $\mathbf{w}_k$ coming from $\mathbf{W}_{jk}$". For the rest of this section, we ommit the superscript from ^{0} for simplicity, but it should be understood that the zeroed version of $\mathbf{W}$ is used by default unless stated otherwise.

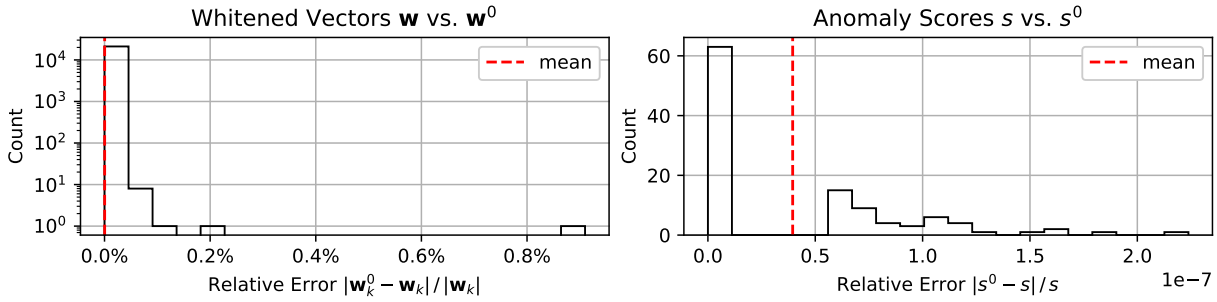


Figure 4.30: Histograms of relative errors between the original and zeroed versions of the whitened vectors and the anomaly scores. The superscript ^{0} in the notation indicates that $\mathbf{w}$ and s are computed from $\mathbf{W}^0$. The modifications in $\mathbf{W}$ do not alter the whitened vector $\mathbf{w}$ significantly, so the anomaly score s also remains practically the same.

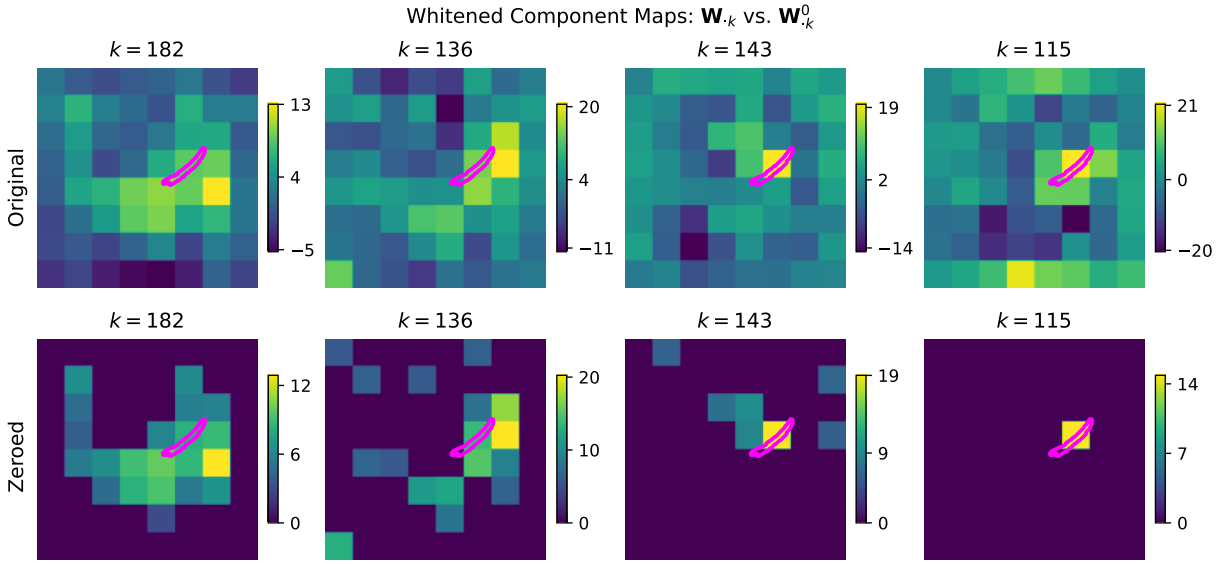


Figure 4.31: Visual effects of the zeroing process on $\mathbf{W}_{\cdot k}$. Examples of $\mathbf{W}_{\cdot k}$ from an image in the Hazelnut dataset from MVTec-AD. The parts of the map that cancel out each other in the average pooling are set to zero, which has an effect of cleaning the signal in each component. The contour of the ground truth anomaly annotation is overlaid in pink for reference.

Contribution Tensor C The partial contributions stored in $\mathbf{c}$ and all $\mathbf{c}^{(k)}$ are combined to build a tensor $\mathbf{C}$. Let $\mathbf{C} \in \mathbb{R}^{M \times d}$, the *contribution tensor*, be defined such that

$$\mathbf{C}_{jk} = \mathbf{c}_k \, \mathbf{c}_j^{(k)} = \left(\frac{\mathbf{w}_k^2}{\sum_{k=1}^{d} \mathbf{w}_k^2} \right) \left(\frac{\mathbf{W}_{jk}}{\sum_{j=1}^{M} \mathbf{W}_{jk}} \right) \quad . \tag{4.36}$$

From this definition, it follows that $\mathbf{C}_{jk} \in [0, 1]$ and $\sum_{k=1}^{d} \sum_{j=1}^{M} \mathbf{C}_{jk} = 1$ (recall that $\mathbf{W}$ is the zeroed version). One can build a tensor $s\,\mathbf{C}$ such that the sum of all its entries equals s, thus a value $\mathbf{C}_{jk}$ can be interpreted as the "percentage of s coming from $\mathbf{W}_{jk}$".

Contribution Map $\overline{\mathbf{C}}$ Finally, an "total contribution" map $\overline{\mathbf{C}}$ is defined by summing the values in $\mathbf{C}$ over the channels. Let $\overline{\mathbf{C}} \in \mathbb{R}^M$, the *contribution map*, be defined such that

$$\overline{\mathbf{C}}_j = \sum_{k=1}^{d} \mathbf{C}_{jk} \quad , \tag{4.37}$$

which satisfies $\sum_{j=1}^{M} \overline{\mathbf{C}}_j = 1$ and $\overline{\mathbf{C}}_j \in [0, 1]$. Therefore, the values $\overline{\mathbf{C}}_j$ can be interpreted as the "percentage of s coming from the pixel at position j". Examples of $\overline{\mathbf{C}}$ are shown in Fig. 4.32.

Notice that, unlike other methods, the heatmaps $\overline{\mathbf{C}}$ *explain* the anomaly score s of the image. In PatchCore and PaDiM, for instance, the models actually generate the heatmaps first, then apply some type of spatial aggregation to obtain the image-wise score.

Excess Contribution After extensively analyzing the heatmaps qualitatively, we noticed that the contributions $\mathbf{C}_{jk}$ that mostly correspond to actual anomalies often concentrate in a few (j, k) positions. Taking advantage of this observation, we propose an extra post-processing step to further enhance the contrast of the heatmaps. The *excess contribution tensor* $\mathbf{C}^{\text{ex}}$ is computed by zeroing out the lowest values in $\mathbf{C}$ and renormalizing the remaining values. The process is similar to the one illustrated in Fig. 4.29, but the threshold τ is set with a different criterion – since all values are positive, there is no zero-crossing point on the cumulative sum. The threshold τ is chosen such that

$$\sum_{j=1}^{M} \sum_{k=1}^{d} \mathbf{C}_{jk} \, \mathbb{1}_{[\mathbf{C}_{jk} > \tau]} = P_{\text{ex}} \quad . \tag{4.38}$$

In simple terms, τ sets a cutoff point such that the retained values account for $P_{\text{ex}} = 15\%$ (by default) of the total score. Intuitively, it was observed that, in the anomalous images, $1 - P_{\text{ex}} = 85\%$ of the total score was spread over irrelevant, normal regions; thus, the *excess* beyond that point is the interesting part. We stress that this is a heuristic choice (averaged over all datasets) that could be adjusted according to the use case.

Let $\mathbf{C}^{\text{ex}} \in [0, 1]^{M \times d}$ be the modified version of $\mathbf{C}$ such that

$$(\mathbf{C}^{\text{ex}})_{jk} = \begin{cases} 0 & \text{if } \mathbf{C}_{jk} < \tau \\ \frac{\mathbf{C}_{jk}}{P_{\text{ex}}} & \text{otherwise} \end{cases} \quad , \tag{4.39}$$

and $\overline{\mathbf{C}^{\text{ex}}} \in [0, 1]^M$ is correspondingly defined such that

$$(\overline{\mathbf{C}^{\text{ex}}})_j = \sum_{k=1}^{d} (\mathbf{C}^{\text{ex}})_{jk} \quad . \tag{4.40}$$

Examples of $\overline{\mathbf{C}^{\text{ex}}}$ are shown in Fig. 4.32. Notice how the contrast of the anomalous regions is further enhanced in these heatmaps. However, it does not necessarily *correct* erroneous detections, which can be seen in the cable and zipper examples.

Segmentation Performance Table 4.4 shows the segmentation performance on the MVTec-AD datasets of different heatmaps in terms of the AUROC score. We consider all the heatmaps mentioned in this section: the Mahalanobis distance map $\mathbf{s}$, the contribution map $\overline{\mathbf{C}}$, and the excess contribution map $\overline{\mathbf{C}^{\text{ex}}}$. Note that the different heatmaps' correspond to the same model. The image-wise AUROC is also shown for reference. Notice that we used a feature extractor with a relatively low performance compared to what was achieved in Section 2.4. It must be emphasized that the methodology presented here does not aim at optimizing the segmentation performance, but rather at providing a qualitative analysis of the image scoring process. The performance of *mdmpa* seems to be unrelated to the detection (*i.e.* image-wise) performance.

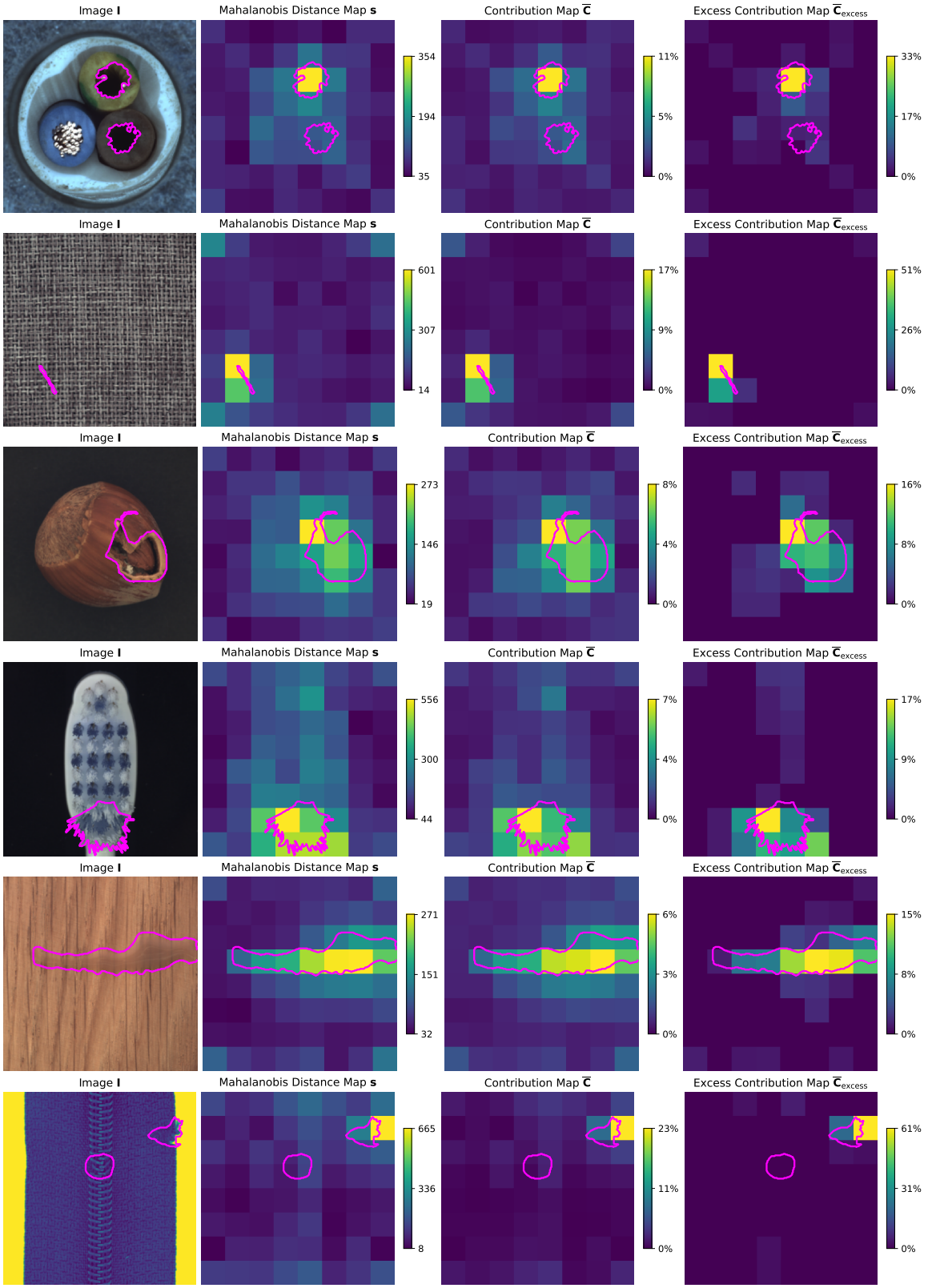


Figure 4.32: Heatmaps from **W** with different strategies.

Table 4.4: Anomaly segmentation performances of EfficientNet-B0 on the MVTec-AD dataset. Image-wise and pixel-wise AUROC scores (higher is better, in %) are reported. Each row corresponds to a single model in a different dataset.

		Pixelwise-wise		
	Image-wise	$\mathbf{a}$	$\mathbf{C}$	$\mathbf{C}^{\text{ex}}$
bottle	95.0	50.8	88.9	85.1
cable	92.5	52.3	83.9	78.6
capsule	86.5	55.2	91.2	85.5
carpet	66.1	50.2	96.8	95.1
grid	58.3	52.6	85.4	79.3
hazelnut	88.8	52.7	92.4	90.6
leather	51.6	50.2	97.8	97.3
metal nut	86.2	52.0	69.7	62.4
pill	94.1	52.6	81.2	78.4
screw	75.0	59.3	88.8	79.1
tile	92.4	50.3	89.2	86.6
toothbrush	58.3	54.5	91.4	87.7
transistor	90.6	51.6	69.3	67.4
wood	71.1	50.2	87.1	84.3
zipper	93.7	52.7	84.0	79.6
mean	80.0	52.5	86.5	82.5

4 Conclusion

This chapter presented a novel dimension reduction strategy for AD with MVG-based models. Using the eigendecomposition of the covariance matrix with a greedy search algorithm, a minimal set of components is selected to yield optimal AD performance. Through extensive experiments, we assessed the effectiveness and generalization capacity of this approach.

In our first experiment, we intentionally overfit the test set, demonstrating that it is possible to achieve high performance with very small embeddings from a subset of eigencomponent projections. Further analyses revealed that the features from the pre-trained model exhibit substantial redundancy and some spurious features that can be detrimental to the model's performance. Our findings show that the per-eigencomponent variance does not correlate with its efficacy in anomaly detection, contradicting existing theories. Besides, a previous belief that deeper layers are less informative for anomaly detection was also contradicted. Our experiments actually suggest that their embeddings are capable of encoding a class's normality in smaller subspaces. The previously seen result – *i.e.* deeper layers are not as informative as the intermediate ones – can be explained by other factors revealed by the visualizations we proposed (*e.g.* resolution mismatch, image border effects, *etc.*).

Subsequent experiments revealed challenges in generalization even when using real anomalies to supervise the selection of eigencomponents. However, the results from Section 2.4 showed that it is possible to better exploit our framework and overcome the difficulties in generalization observed in Sections 2.2.3 and 2.2.4. For instance, the use of a different criterion to select eigen-

components (*i.e.* a different g in Chapter 4) provided better generalization capacity. Even using only synthetic anomalies, our boosted training pipeline consistently improved the performance of the baseline model.

Our conclusions are primarily based on the EfficientNet architecture because it is widely used in this domain, and its performance is consistent. Future research could explore other architectures to further validate and enhance our results. However, the experiments were conducted across multiple datasets and nodes, which gives credibility to the generalizability of our observations.

Using models based on [Rippel, 2021c; Rippel, 2021a; Lin, 2022; Kim, 2021], our results did not achieve SOTA performance. However consistent improvements – compared to the baseline vanilla MVG model – were observed. Other Gaussian-based models, such as Neighboring Pixel-based industrial Anomaly Detection (N-pad) [Jang, 2023] and Focus Your Distribution (FYD) [Zheng, 2022] (both based on PaDiM [Defard, 2021]), have shown results inline with SOTA performance. Integrating the insights from these both works could be a promising direction to achieve better performance and robustness in VAD.

Our framework could also be extended to cover GMM-based models. This would enable better modeling of multi-mode distributions (*e.g.* Fig. 4.21), which are common in real-world data. However, the adaptation of the greedy search algorithm to GMM models is not straightforward, so further research is needed to address this issue.

In summary, this work extends the findings of [Rippel, 2021c; Rippel, 2021a] regarding dimensionality reduction in MVG models, contributing to a deeper understanding of the feature spaces learned by CNNs and their applicability for AD. While the proposed method does not yet reach SOTA performance, it offers a simple alternative with negligible computational costs relative to a CNN's inference, and it can be integrated seamlessly without requiring retraining. Additionally, the visualization techniques introduced in Section 3 enhance interpretability, providing valuable insights into the model's decision-making process.

Chapter 5

One-class classification: adapting the hypersphere classifier to segmentation

Abstract

In this chapter, we propose an incremental improvement to FCDD [Liznerski, 2021]. FCDD is trained using a one-class classification (OCC) approach from the feature map outputed by a backbone CNN. We reformulate its SVDD-inspired loss to make better use of pixel-level annotation provided by synthetic anomalies or a few real anomalies. Our formulation consists of applying a loss function to the pixels then averaging it over the entire batch of pixels. We demonstrate consistent improvement in performance across datasets with less training epochs. Two preprints relative to this work have been published [Bertoldo, 2022; Bertoldo, 2023d].

Résumé

Dans ce chapitre, nous proposons une amélioration incrémentale de FCDD [Liznerski, 2021]. La FCDD est entraînée à l'aide d'une approche de classification de une classe à partir de la carte de *features* produite par un réseau de neurones convolutif de base. Nous reformulons sa perte inspirée du Support Vector Data Description (SVDD) afin de mieux utiliser l'annotation au niveau des pixels fournie par des anomalies synthétiques ou quelques anomalies réelles. Notre formulation consiste à appliquer une fonction de perte aux pixels et à en faire la moyenne sur l'ensemble du lot de pixels. Nous démontrons une amélioration des performances consistante sur des ensembles de données avec moins de *epoch* d'entraînement. Deux prépublications relatives à ce travail ont été publiées [Bertoldo, 2022; Bertoldo, 2023d].

Contents

1 Background

In this chapter, we explore an incremental improvement to a one-class classification model: the Fully Convolutional Data Descriptor (FCDD) model [Liznerski, 2021], which is – in fact – a one-class *segmentation* model. One-class classification is a model family where the goal is to learn a mapping from the input space to a feature space where the normal instances are grouped together and the anomalous instances are separated from them. This approach may look similar to the one presented in Chapter 4, but the key difference is that (here) the parameters of the feature extractor are learned. This section summarizes the predecessors that led to the formulation presented in FCDD [Liznerski, 2021] focusing on their incremental changes. For the sake of clarity, some details, such as regularization terms, have been omitted. An extensive review of OCC models can be found in [Perera, 2021].

OCC was first formulated as an adaptation of Support Vector Machine (SVM), which learns to separate anomalous data points from the origin with a hyperplane [Schölkopf, 2001] or a hypersphere [Tax, 2004] in the feature space. The latter, SVDD [Tax, 2004], minimizes the radius of a hypersphere by incentivizing – via slack variables – the normal/anomalous data points to fall inside/outside the hypersphere. In other words, normal and anomalous instances are penalized in opposite directions. Notice that this formulation assumes supervision, and a kernel version in its dual form is known to considerably improve the model's expressive power.

One-Class Deep SVDD [Ruff, 2018], a deep-learning model inspired by SVDD, uses a neural network $\phi_\omega : \mathbb{R}^n \mapsto \mathbb{R}^d$, with parameters ω, to extract features vectors from normal images $\mathbf{I} \in \mathbb{R}^n$ (spatial and channel dimensions simplified in a single dimension). The model's parameters are optimized to minimize the distance of the feature vectors to the center of a hypersphere in the feature space:

$$\frac{1}{N}\sum_{i=1}^{N} \| \phi_\omega(\mathbf{I}_i) - \mathbf{c} \|_2^2 \quad , \tag{5.1}$$

where $\mathbf{c} \in \mathbb{R}^d$ is the center of the hypersphere in the feature space and N is the number of instances in the batch. The distance to this center is then used as anomaly score at inference time. Notice that this version does *not* consider anomalous instances, which makes the training less stable. The objective function in Eq. (5.1) often leads to feature collapse (ϕ_ω converging to a constant function), so the network needs to be meaningfully regularized.

Among several strategies proposed to prevent feature collapse, [Ruff, 2020; Ruff, 2021b] proposed an extension of Eq. (5.1) that brings back a term to include anomalous samples. Since

the original term in Eq. (5.1) *pulls* normal instances towards the center, they proposed a term that *pushes* the anomalous instances away from it. Following this logic, we define the *push function* as

$$p : s \;\mapsto\; -\log(1-\exp(-s)) \quad \forall s \in \mathbb{R}_+ \quad , \tag{5.2}$$

which is employed to build the hypersphere classifier (HSC) loss function:

$$\frac{1}{N}\sum_{i=1}^{N} \; (1-l_i) \;\; \| \, \phi_\omega(\mathbf{I}_i) - \mathbf{c} \, \|_2^2 \;\; + \;\; l_i \;\; p\left(\| \, \phi_\omega(\mathbf{I}_i) - \mathbf{c} \, \|_2^2 \right) \quad , \tag{5.3}$$

where $l_i \in \{0, 1\}$ is the label of the image $\mathbf{I}_i$ such that "0" and "1" mean "normal" and "anomalous" respectively. Notice that this makes the two terms in the sum multually exclusive; for a single instance, only the pull *or* only the push term contributes to the loss. The authors also noticed that, due to the squared Euclidean distance in Eq. (5.3), the loss tends to be dominated by a few instances. As a counter measure, it was replaced by the Pseudo-Huber loss function

$$h : \mathbf{z} \;\mapsto\; \sqrt{\| \, \mathbf{z} \, \|_2^2 \;+\; 1} \;\; - \;\; 1 \quad \forall \, \mathbf{z} \in \mathbb{R}^d \quad , \tag{5.4}$$

which is more robust to outliers. The final HSC loss function is then

$$\frac{1}{N}\sum_{i=1}^{N} \; (1-l_i) \;\; h\left(\phi_\omega(\mathbf{I}_i) - \mathbf{c}\right) \;\; + \;\; l_i \;\; p\left(h\left(\phi_\omega(\mathbf{I}_i) - \mathbf{c}\right)\right) \quad . \tag{5.5}$$

In simple terms, by minimizing Eq. (5.5), the network learns parameters ω that map normal instances close to the center of the hypersphere and anomalous instances far from it.

FCDD, illustrated in Fig. 5.1, adapted the HSC loss to a patch-wise anomaly detection (*i.e.* anomaly segmentation/localization). A backbone convolutional neural network extracts a feature map – generally in a lower resolution – from the input image. In this 3D tensor, each spatial position is a feature vector encoding information from a local patch of the image. Then, an extra 1x1 convolution is added on top of backbone to embed a learnable center $\mathbf{c}$ in the latent space.

At training time, the output of the 1x1 convolution is interpreted as distance – *i.e.* analogous to "$\phi_\omega(\mathbf{I}_i) - \mathbf{c}$" in Eq. (5.5) – and used to compute the HSC loss. At inference time, an anomaly score map is generated from these patch-wise distances, which are upsampled to the input resolution. In the next section we analyze the loss function used in FCDD in detail and propose modifications to improve its anomaly segmentation performance.

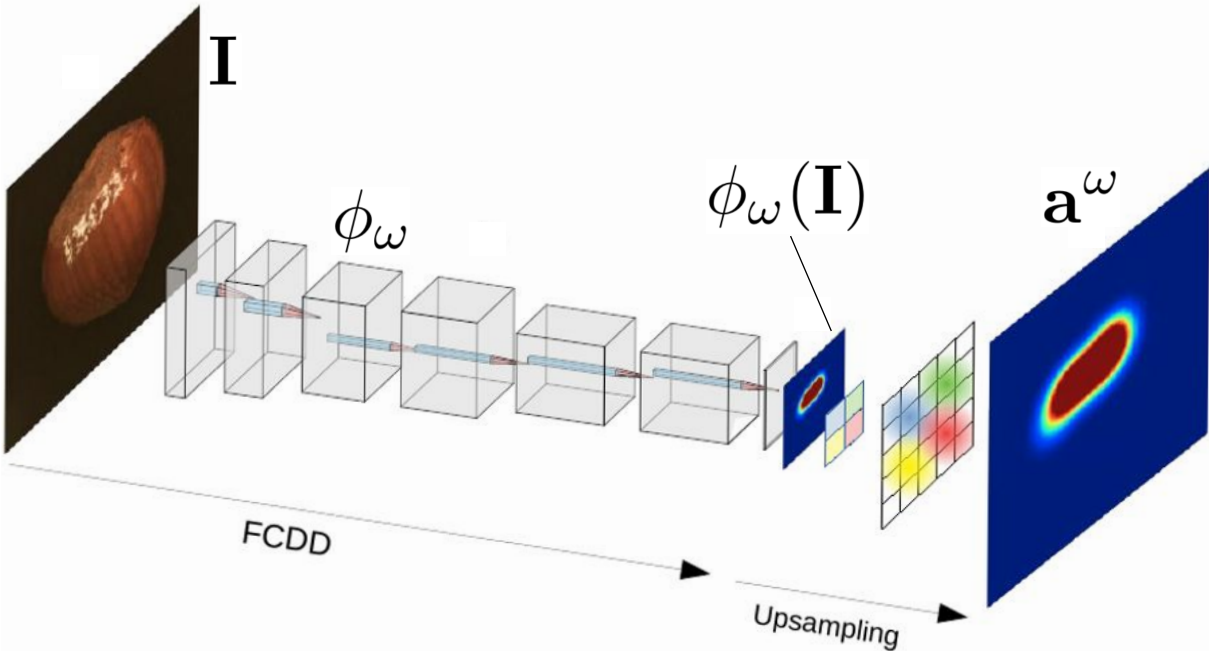


Figure 5.1: Fully Convolutional Data Descriptor (FCDD). A pre-trained (and fine-tuned) convolutional neural network without fully connected layers is combined with an extra 1x1 convolution layer to generate locality-preserving feature vectors. In the last (low resolution) layer, the distances to the hypersphere's center are upsampled to the network's input resolution and used as an anomaly score heatmap. Adapted from [Liznerski, 2021].

2 Loss Function

Let $\mathbf{I}$ be an image with M pixels (spatial dimensions confounded) and $\mathbf{y} \in \{0,1\}^M$ its ground truth anomaly mask, where "0"/"1" means "normal"/"anomalous" pixel respectively. Given a batch of N images and annotations masks, $\mathbf{y}_{ij}$ refers to the (pixel-level) label at the spatial position $j \in [\![M]\!]$ in the image of index $i \in [\![N]\!]$ in the batch, where $[\![n]\!]$ denotes the set $\{1,\dots,n\}$ $\forall n \in \mathbb{N}$.

An input image is passed to a backbone CNN with parameters ω to extract a feature map with low-resolution $M < M$, then a 1×1 convolution with bias outputs a single value per pixel. Let $\phi_\omega(\mathbf{I}) \in \mathbb{R}^M$ note the output of this operation – *i.e.* including the backbone *and* the 1x1 convolution. Finally, the huber function h and a Gaussian kernel-based upsampling are applied to obtain an anomaly score map

$$\mathbf{a}^\omega = \texttt{upsample}(h(\phi_\omega(\mathbf{I}))) \in \mathbb{R}_+^M \quad , \tag{5.6}$$

where h is applied pixel-wise (at each spatial position independently) and the superscript "ω" reminds that the anomaly scores directly depend on the network's parameters. Notice that the center $\mathbf{c}$ has been omitted as it is implicitly embedded in the 1×1 convolution. The loss function

used in [Liznerski, 2021] can be written as

$$\frac{1}{N}\sum_{i=1}^{N}\left[\left(\frac{1}{M}\sum_{j=1}^{M}(1-\mathbf{y}_{ij})\,\mathbf{a}_{ij}^{\omega}\right) \;+\; p\left(\frac{1}{M}\sum_{j=1}^{M}\mathbf{y}_{ij}\,\mathbf{a}_{ij}^{\omega}\right)\right]. \tag{Baseline}$$

Eq. (Baseline) can be read as the cross-image average ($\frac{1}{N}\sum_{i=1}^{N}$) of two mutually exclusive terms that pull or push feature vectors – as in the image-wise HSC presented in Section 1. The difference is that the anomaly scores – associated with the notion of distance from the center of a hypersphere – are pixel-wise and aggregated over the spatial dimensions.

Note that the spatial aggregation *looks* like a cross-pixel (within-image) average ($\frac{1}{M}\sum_{j=1}^{M}$), but they are not. In fact, $\mathbf{y}_{ij}$ zeroes part of the iterands, so each of the the two sum $\sum_{j=1}^{M}$ accounts for less than M terms. By correcting that, one would have a minimization of the average distance of the normal pixels and a maximization of the average distance of the anomalous pixels – where p is responsible for inverting the direction of the gradient for the anomalous pixels. This design makes the backpropagation focus on very high-scored normal pixels and very low-scored anomalous pixels.

We propose, as a substitute to Eq. (Baseline), the loss function

$$\frac{1}{N\,M}\sum_{i=1}^{N}\sum_{j=1}^{M}(1-\mathbf{y}_{ij})\,\mathbf{a}_{ij}^{\omega} \;+\; \lambda\,\mathbf{y}_{ij}\,p\left(\mathbf{a}_{ij}^{\omega}\right) \quad , \tag{Ours}$$

where λ is a fixed parameter set to the ratio between the number of normal pixels over the number of anomalous pixels in the batch. Our version applies p on each pixel individually and a cross-image, cross-pixel average is computed – *i.e.* all pixels in the batch are equally weighted. The balancing factor λ compensates for the imbalance of normal and anomalous pixels. Other – more advanced – techniques exist but we deliberately do not focus on that topic because our tests showed that, while simple, this balancing is effective.

3 Experiments

Procedure We tested our proposed loss function with the same procedure as in [Liznerski, 2021] and compare it on MVTec-AD – each dataset being trained and tested independently. A VGG-11 backbone, pre-trained on ImageNet, frozen up to the third max-pooling and cut before the forth max-pooling is used; the network is fine-tuned using stochastic gradient descent (SGD) with a batch size of 128 images, Nesterov momentum of 0.9, weight decay of 10^{-4}, and initial learning rate of 10^{-3} decreasing by a factor of 1.5% at each epoch. One epoch is defined as 10 iterations over the training set with each image having 50% chance to be replaced by an anomalous image. Each experiment was repeated six times with different seeds, and the reported performances are cross-seed averages.

Preprocessing At training time, images are pre-processed and augmented with the following sequence of operations: resize to 240 x 240; with 50% chance each, resize to 224 x 224 or crop to

Figure 5.2: Synthetic anomalies generated with confetti noise.

224 x 224 in a random position; apply color jitter with all parameters set to 0.04 or all parameters set to 0.0005 with 50% chance each; apply Gaussian noise with zero mean and standard deviation set to 10% of the image's pixel values' standard deviation (all channels confounded) with 50% chance on each pixel; apply local contrast normalization (all channels confounded); normalize the data between 0 and 1 (min-max normalization). At inference time, images are resized to 224 x 224 then the last two last two steps from the training pipeline are applied (contrast and normalization).

Anomaly Source As in [Liznerski, 2021], we consider two sources of anomalies: synthetic and real. Synthetic anomalies, referred to as "*unsupervised* setting", are automatically generated by superposing a confetti noise over a normal image (any modified pixel is considered anomalous). The confetti noise consist of randomly sized, placed, and colored squares (see Fig. 5.2). This image corruption occurs before the data preprocessing/augmentation described above. Real anomalies, referred to as "*semi-supervised* setting", are removed from the test set and added to the training set. Only one image of each anomaly type is transferred, and the number of anomaly types in MVTec-AD's dataset ranges between 1 and 7 – so the number of anomalies in the training varies accordingly. Our focus is on the unsupervised setting but the semi-supervised setting ("a few" anomalies available) is also tested for reference.

Modifications We run the gradient descent for only 50 epochs instead of 200, and the synthetic anomalies are not smoothed on the borders to avoid label uncertainty.

Hardware and Software We ran our experiments on cluster nodes PowerEdge C4130 (processor Xeon E5-2680 V4, 7 cores, 2.4GHz), each with four GPUs Tesla P100-SXM2 (16 GiB). The minimum single-GPU memory requirement is 5.6 GiB, and the (category-specific) average run time spans from 33 minutes to 2 hours. We used the code publicly provided by [Liznerski, 2021] with modifications to integrate the losses described in Section 2.

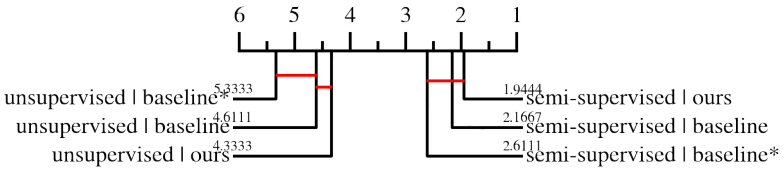
4 Results

Average Performance Tables 5.1 and 5.2 compare the cross-category average results obtained with the baseline loss from [Liznerski, 2021] (Eq. (Baseline)) and ours (Eq. (Ours)) in terms of, respectively, area under the receiver operating characteristic curve (AUROC) and area under the precision-recall (AUPR) – definitions can be found in Chapter 2. We report the cross-dataset average scores and their standard deviations (in parenthesis). Under the same supervision setting, the best performance is in bold. The baseline performances in terms of AUROC are also copied from [Liznerski, 2021] (with "$*$").

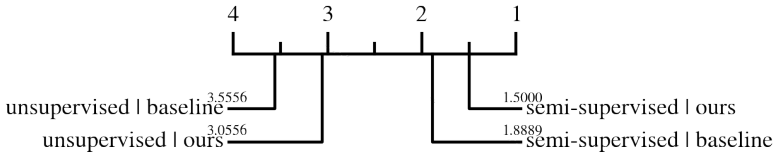
Critical Difference Diagrams Fig. 5.3 shows critical difference (CD) diagrams using the pairwise Wilcoxon signed-rank test with Bonferroni-Holm correction with a (nominal) significance level of $\alpha = 10\%$ (probability of a type I error) considering the two-sided alternative hypothesis. This statistical test's null hypothesis is that two methods are equivalent thus expected to rank equally on average. It makes non-parametrical comparisons across datasets, which is suitable because the datasets are independent problems so they are not necessarily comparable. It provides a stronger statistical argument than comparing average performances because rejecting its null hypothesis means that a method is consistently better than another across different problems. Methods connected by a red horizontal bar in Fig. 5.3 are not significantly different according to the pairwise tests.

5 Discussion and Perspectives

Our proposed modifications provided, on average, an increase in performance over object, texture, and all datasets confounded. Both metrics (AUROC and AUPR) increased with our approach on both settings, and the variance decreased in the unsupervised setting (*cf*. standard deviations in parenthesis). Fig. 5.3 confirms that our loss function is ranked better than the previous one, with consistently better AUPR across different datasets. Finally, we (re-)observe that semi-supervised setting generally performs better than the unsupervised one even having very few anomalous images (1 up to 7). This suggests that disposing of real (or realistic) anomalies is a major factor for the HSC model. While consistent, we presented an incremental improvement, and we believe that further research could lead to more substantial improvements by exploring segementation-specific architectures like U-Net [Ronneberger, 2015b].



(a) Pixel-wise AUROC



(b) Pixel-wise Average Precision

Figure 5.3: Critical difference diagrams from the category-specific scores on the MVTec AD dataset using the Wilcoxon signed-rank test with the Bonferroni-Holm correction with a confidence level of $\alpha = 10\%$ at for each pair of methods. Values on the scale are average rankings, and the methods within a same horizontal bar (a clique) are *not* significantly different according to the pair-wise statistical test. The scores of the methods marked with "*" are from [Liznerski, 2021].

Table 5.1: Pixel-wise AUROC (in percentage) on the MVTec-AD's datasets.

	semi-supervised		
	baseline*	baseline	ours
objects	95.2*	95.7 (1.4)	**95.8 (1.4)**
textures	97.0*	97.8 (0.4)	**97.9 (0.4)**
all	95.8*	96.4 (1.0)	**96.6 (1.0)**
	unsupervised		
	baseline*	baseline	ours
objects	91.0*	**92.5 (1.1)**	92.5 (1.0)
textures	92.8*	92.8 (1.3)	**94.1 (0.7)**
all	91.6*	92.6 (1.1)	**93.1 (0.9)**

Table 5.2: Pixel-wise AUPR (in percentage) on the MVTec-AD's datasets.

	semi-supervised		unsupervised	
	baseline	ours	baseline	ours
objects	52.0 (5.8)	**53.4 (5.8)**	36.6 (4.3)	**38.7 (2.3)**
textures	50.3 (4.5)	**54.7 (4.1)**	37.8 (2.8)	**40.2 (2.0)**
all	51.4 (5.3)	**53.8 (5.2)**	37.0 (3.8)	**39.2 (2.2)**

Conclusion

This thesis contributes to the field of visual anomaly detection with novel ideas that span from evaluation and practical applicability to model-specific improvements with a particular focus on Gaussian-based approaches.

Motivated by the large amount of research focusing on models and recent advances in the field, evaluation has been neglected. This thesis aims to address this by proposing a more challenging and realistic evaluation metric with AUPIMO to encourage the development of more reliable models. We believe AUPIMO can be a key contribution to future research in the field of anomaly detection. It relies on a more rigorous model validation embedded in the evaluation metric, which focuses on minimizing false positives on normal samples. This rather simple but harsh condition imposes a more reliable operating point to the models. We believe AUPIMO has the potential to become a standard evaluation and encourage a higher degree of trustworthiness, thus bridging the gap between research and real-world scenarios.

Beyond evaluation, this thesis also addresses the practical usability of anomaly detection models. While a lot of work has been done to improve the performance of anomaly detection models, the final usability of these models is often overlooked. The models' unsupervised nature is often broken by anomaly-dependent statistics-based threshold selection, which is the case for heatmap normalization. Recognizing the limitations of conventional color-coded heatmaps for anomaly localization, this thesis introduces an alternative post-processing idea. The proposed threshold selection method leverages image-specific information only. We show that avoiding the use of a global threshold can lead to significant improvements in the visual quality of anomaly localization.

The in-depth analysis of Gaussian-based models challenges some prevailing notions in the field. Previous research suggested that low-variance subspaces are more discriminative for anomaly detection, but our analysis shows that there is no clear correlation between the variance and the model performance. Moreover, deeper layers were believed to be less discriminative due to bias towards the classes from the pre-training task. Our analysis shows that deeper layers systematically have more discriminative power than the shallower ones, but they also carry more irrelevant information and resolution-related issues. The greedy optimization of principal components introduced in this thesis leads to a novel feature reduction method that consistently improves performance by removing irrelevant components. Additionally, the introduction of a visualization tool for image-wise Gaussian models enables anomaly localization, which was

previously not possible.

The contributions of this thesis extend beyond the specific algorithms and analyses presented. By addressing fundamental aspects of evaluation, interpretability, and model optimization, this work has broader implications for the field of visual anomaly detection and its applications. The emphasis on rigorous evaluation and unsupervised techniques fosters greater confidence in the reliability and trustworthiness of anomaly detection models. This, in turn, can lead to wider adoption and integration of these models in critical real-world applications.

Future research can build upon this work by exploring the adaptability of AUPIMO to other domains, including video anomaly detection, 3D images, and point clouds. Incorporating domain-specific considerations into the validation criterion will be crucial for ensuring the effectiveness of AUPIMO in diverse contexts. Additionally, further investigation into unsupervised threshold selection methods can lead to more robust and reliable anomaly localization techniques. This could involve combining morphological operations or a contrario-like techniques to further refine anomaly segmentation results.

In the realm of Gaussian-based models, future work can focus on extending the analysis to jointly optimize subspaces from multiple layers and exploring the integration of the proposed subspace-based feature reduction method with other Gaussian-based models. This could unlock further performance gains and enhance the applicability of these models in complex anomaly detection tasks.

By addressing fundamental challenges and opening new avenues for research, this thesis contributes to the advancement of visual anomaly detection and paves the way for the development of more robust, reliable, and interpretable anomaly detection systems. These advancements have the potential to significantly impact various industries and applications, enabling more efficient and effective anomaly detection in diverse real-world scenarios.

Publications

AUPIMO was developed during Google Summer of Code (GSoC) 2023 at OpenVINO (Intel). Project page: https://summerofcode.withgoogle.com/archive/2023/projects/SPMopugd.

International Conferences

1. Tetiana Gula and **João P. C. Bertoldo**, "Gaussian Image Anomaly Detection with Greedy Eigencomponent Selection, International Conference on Computer Vision (ICCV)", 2023

2. **João P. C. Bertoldo** and David Arrustico, "Visualization for Multivariate Gaussian Anomaly Detection in Images, Twelfth International Conference on Image Processing Theory, Tools and Applications (IPTA)", 2023

3. Zeyu Jiang, **João P. C. Bertoldo**, and Etienne Decencière, "Heuristic Hyperparameter Choice for Image Anomaly Detection, Twelfth International Conference on Image Processing Theory, Tools and Applications (IPTA)", 2023

4. **João P. C. Bertoldo**, Dick Ameln, Ashwin Vaidya, and Samet Akçay, "AUPIMO: Redefining Visual Anomaly Detection Benchmarks with High Speed and Low Tolerance", The 35th British Machine Vision Conference (BMVC), 2024

Preprints

1. **João P. C. Bertoldo** and Etienne Decencière, "[Reproducibility Report] Explainable Deep One-Class Classification", arXiv, 2022

2. **João P. C. Bertoldo**, Santiago Velasco-Forero, Jesus Angulo, and Etienne Decencière, "Adapting the Hypersphere Loss Function from Anomaly Detection to Anomaly Segmentation", arXiv, 2023

Public available code

AUPIMO is available in a standalone repository and integrated in Anomalib (released in version `1.2.0`) as a submodule. Tutorials on how to use AUPIMO are available in Anomalib's repository.
Standalone: https://github.com/jpcbertoldo/aupimo
Anomalib: https://github.com/openvinotoolkit/anomalib

The threshold selection method from Chapter 3 is available in AUPIMO's repository in a different branch.
https://github.com/jpcbertoldo/aupimo/tree/jpcbertoldo/per-image-threshs

The visualization tool for image-wise Gaussian-based models from Section 3.2 (Chapter 4) is available in a standalone repository.
https://github.com/jpcbertoldo/gaussian-ad-viz

Bibliography

[Achanta, 2012] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods". *IEEE Trans. Pattern Anal. Mach. Intell.* 34.11 (2012), pp. 2274–2282 (cit. on p. 53).

[Agarwal, 2006] Pankaj K Agarwal, Sariel Har-Peled, and Kasturi R Varadarajan. "Geometric approximation via coresets survey". *Current Trends in Combinatorial and Computational Geometry, E. Welzl, ed., Cambridge University Press, Cambridge* (2006) (cit. on p. 8).

[Akcay, 2022] Samet Akcay, Dick Ameln, Ashwin Vaidya, Barath Lakshmanan, Nilesh Ahuja, and Utku Genc. "Anomalib: A Deep Learning Library for Anomaly Detection". *IEEE Int. Conf. Image Process.* 2022, pp. 1706–1710 (cit. on pp. 25, 31).

[Akcay, 2019] Samet Akcay, Amir Atapour-Abarghouei, and Toby P. Breckon. "GANomaly: Semi-supervised Anomaly Detection via Adversarial Training". *Asian Conf. Comput. Vis.* Ed. by C. V. Jawahar, Hongdong Li, Greg Mori, and Konrad Schindler. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 622–637 (cit. on p. 11).

[Akçay, 2019] Samet Akçay, Amir Atapour-Abarghouei, and Toby P. Breckon. "Skip-GANomaly: Skip Connected and Adversarially Trained Encoder-Decoder Anomaly Detection". *2019 International Joint Conference on Neural Networks (IJCNN)*. 2019 International Joint Conference on Neural Networks (IJCNN). 2019, pp. 1–8 (cit. on p. 11).

[Astrid, 2021] Marcella Astrid, Muhammad Zaigham Zaheer, Jae-Yeong Lee, and Seung-Ik Lee. "Learning Not to Reconstruct Anomalies". *Brit. Mach. Vis. Conf.* 2021 (cit. on p. 10).

[Bae, 2022] Jaehyeok Bae, Jae-Han Lee, and Seyun Kim. *Image Anomaly Detection and Localization with Position and Neighborhood Information*. 2022 (cit. on p. 8).

[Bai, 2023] Haoping Bai, Shancong Mou, Tatiana Likhomanenko, Ramazan Gokberk Cinbis, Oncel Tuzel, Ping Huang, et al. *VISION Datasets: A Benchmark for Vision-based InduStrial InspectiON*. 2023 (cit. on p. 5).

[Bao, 2023] Jinan Bao, Hanshi Sun, Hanqiu Deng, Yinsheng He, Zhaoxiang Zhang, and Xingyu Li. *BMAD: Benchmarks for Medical Anomaly Detection*. 2023 (cit. on p. 5).

[Battikh, 2022] Muhammad S. Battikh and Artem A. Lenskiy. "Latent-Insensitive Autoencoders for Anomaly Detection". *Mathematics* 10.1 (2022) (cit. on p. 10).

[Batzner, 2024] Kilian Batzner, Lars Heckler, and Rebecca König. "EfficientAD: Accurate Visual Anomaly Detection at Millisecond-Level Latencies". *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2024, pp. 128–138 (cit. on pp. 11, 31, 37, 44, 60, 97).

[Benavoli, 2016] Alessio Benavoli, Giorgio Corani, and Francesca Mangili. "Should We Really Use Post-Hoc Tests Based on Mean-Ranks?" *J. Mach. Learn. Res.* 17.5 (2016), pp. 1–10 (cit. on p. 33).

[Bergman, 2020] Liron Bergman, Niv Cohen, and Yedid Hoshen. *Deep Nearest Neighbor Anomaly Detection*. 2020. arXiv: 2002.10445[cs,stat] (cit. on p. 8).

[Bergmann, 2021a] 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". *Int. J. Comput. Vis.* 129.4 (2021), pp. 1038–1059 (cit. on pp. 5, 14, 19, 31, 75).

[Bergmann, 2022] Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, and Carsten Steger. "Beyond Dents and Scratches: Logical Constraints in Unsupervised Anomaly Detection and Localization". *Int. J. Comput. Vis.* 130.4 (2022), pp. 947–969 (cit. on pp. 5, 42).

[Bergmann, 2019] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. "MVTec AD - A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection". *IEEE Conf. Comput. Vis. Pattern Recog.* 2019, pp. 9592–9600 (cit. on pp. 5, 14, 17, 75).

[Bergmann, 2020] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. "Uninformed Students: Student-Teacher Anomaly Detection With Discriminative Latent Embeddings". *IEEE Conf. Comput. Vis. Pattern Recog.* 2020, pp. 4183–4192 (cit. on pp. 11, 44).

[Bergmann, 2021b] Paul Bergmann, Xin Jin, David Sattlegger, and Carsten Steger. *The MVTec 3D-AD Dataset for Unsupervised 3D Anomaly Detection and Localization*. 2021. arXiv: 2112.09045 (cit. on p. 5).

[Bertoldo, 2023a] Joao P C Bertoldo and David Arrustico. *Pixel-wise Gaussian AD visualization (ResNet18, MVTecAD)*. Version 0.0.0. Zenodo, 2023 (cit. on pp. 79, 101).

[Bertoldo, 2023b] João P C Bertoldo and David Arrustico. "Visualization for Multivariate Gaussian Anomaly Detection in Images". *2023 Twelfth International Conference on Image Processing Theory, Tools and Applications (IPTA)*. 2023, pp. 1–6 (cit. on p. 8).

[Bertoldo, 2023c] João P C Bertoldo and David Arrustico. "Visualization for Multivariate Gaussian Anomaly Detection in Images". *2023 Twelfth International Conference on Image Processing Theory, Tools and Applications (IPTA)*. 2023, pp. 1–6 (cit. on pp. 81, 100).

[Bertoldo, 2024] Joao P. C. Bertoldo, Dick Ameln, Ashwin Vaidya, and Samet Akçay. "AUPIMO: Redefining Visual Anomaly Detection Benchmarks with High Speed and Low Tolerance". *Accepted to The 35th British Machine Vision Conference (BMVC 2024)*. 2024 (cit. on pp. 13, 33).

[Bertoldo, 2022] Joao P. C. Bertoldo and Etienne Decencière. *[Reproducibility Report] Explainable Deep One-Class Classification*. 2022 (cit. on p. 122).

[Bertoldo, 2023d] Joao P. C. Bertoldo, Santiago Velasco-Forero, Jesus Angulo, and Etienne Decencière. *Adapting the Hypersphere Loss Function from Anomaly Detection to Anomaly Segmentation*. 2023 (cit. on p. 122).

[Beucher, 2018] Serge Beucher and Fernand Meyer. "The morphological approach to segmentation: the watershed transformation". *Mathematical morphology in image processing*. CRC Press, 2018, pp. 433–481 (cit. on p. 53).

[Blum, 2021] Hermann Blum, Paul-Edouard Sarlin, Juan Nieto, Roland Siegwart, and Cesar Cadena. "The Fishyscapes Benchmark: Measuring Blind Spots in Semantic Segmentation". *International Journal of Computer Vision* 129.11 (2021), pp. 3119–3135 (cit. on p. 5).

[Božič, 2021] Jakob Božič, Domen Tabernik, and Danijel Skočaj. "Mixed supervision for surface-defect detection: From weakly to fully supervised learning". *Computers in Industry* 129 (2021), p. 103459 (cit. on pp. 5, 14).

[Bradski, 2000] G. Bradski. "The OpenCV Library". *Dr. Dobb's Journal of Software Tools* (2000) (cit. on p. 25).

[Cao, 2022] Yunkang Cao, Qian Wan, Weiming Shen, and Liang Gao. "Informative knowledge distillation for image anomaly segmentation". *Knowledge-Based Systems* 248 (2022), p. 108846 (cit. on p. 11).

[Chao, 2022] Huang Chao. "Pixel-Level Feature Clustering Learning for Image Anomaly Detection and Localization". *2022 19th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*. 2022 19th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP). 2022, pp. 1–4 (cit. on p. 11).

[Cimpoi, 2014] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. "Describing Textures in the Wild". *IEEE Conf. Comput. Vis. Pattern Recog.* 2014, pp. 3606–3613 (cit. on p. 90).

[Cohen, 2020] Niv Cohen and Yedid Hoshen. *Sub-Image Anomaly Detection with Deep Pyramid Correspondences*. 2020 (cit. on p. 8).

[Cordier, 2022] Antoine Cordier, Benjamin Missaoui, and Pierre Gutierrez. "Data refinement for fully unsupervised visual inspection using pre-trained networks". *ArXiv* (2022) (cit. on p. 6).

[Cubuk, 2020] Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. "RandAugment: Practical Automated Data Augmentation with a Reduced Search Space". *Adv. Neural Inform. Process. Syst.* Vol. 33. 2020, pp. 18613–18624 (cit. on p. 90).

[Cui, 2023] Yajie Cui, Zhaoxiang Liu, and Shiguo Lian. "A Survey on Unsupervised Anomaly Detection Algorithms for Industrial Images". *IEEE Access* 11 (2023), pp. 55297–55315 (cit. on p. 6).

[Defard, 2021] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. "PaDiM: A Patch Distribution Modeling Framework for Anomaly Detection and Localization". *Int. Conf. Pattern Recog.* Ed. by Alberto Del Bimbo, Rita Cucchiara, Stan Sclaroff, Giovanni Maria Farinella, Tao Mei, Marco Bertini, et al. Springer International Publishing, 2021, pp. 475–489 (cit. on pp. 7, 31, 44, 65, 68, 74, 78, 100, 103, 104, 121).

[Dehaene, 2020] David Dehaene, Oriel Frigo, Sébastien Combrexelle, and Pierre Eline. "Iterative energy-based projection on a normal data manifold for anomaly localization". ICLR. 2020, p. 17 (cit. on p. 8).

[Demšar, 2006] Janez Demšar. "Statistical Comparisons of Classifiers over Multiple Data Sets". *J. Mach. Learn. Res.* 7.1 (2006), pp. 1–30 (cit. on p. 33).

[Deng, 2022] Hanqiu Deng and Xingyu Li. "Anomaly Detection via Reverse Distillation From One-Class Embedding". *IEEE Conf. Comput. Vis. Pattern Recog.* 2022, pp. 9737–9746 (cit. on pp. 11, 44).

[Deng, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "Imagenet: A large-scale hierarchical image database". *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255 (cit. on pp. 5, 7).

[Desolneux, 2008] Agnés Desolneux, Lionel Moisan, and Jean-Michel Morel. *From Gestalt Theory to Image Analysis: A Probabilistic Approach*. Red. by S. S. Antman, L. Sirovich, J. E. Marsden, and S. Wiggins. Vol. 34. Interdisciplinary Applied Mathematics. New York, NY: Springer New York, 2008 (cit. on p. 45).

[Dini, 2022] Afshin Dini and Esa Rahtu. "TPSAD: Learning to Detect and Localize Anomalies With Thin Plate Spline Transformation". *Int. Conf. Pattern Recog.* 2022, pp. 4744–4750 (cit. on p. 8).

[Du, 2022] Xuefeng Du, Zhaoning Wang, Mu Cai, and Yixuan Li. "VOS: Learning What You Don't Know by Virtual Outlier Synthesis". *Int. Conf. Learn. Represent.* 2022 (cit. on p. 65).

[Etherington, 2021] Thomas R. Etherington. "Mahalanobis distances for ecological niche modelling and outlier detection: implications of sample size, error, and bias for selecting and parameterising a multivariate location and scatter method". *PeerJ* 9 (2021), e11436 (cit. on pp. 3, 7).

[Fawcett, 2006] Tom Fawcett. "An introduction to ROC analysis". *Pattern Recognition Letters* 27.8 (2006), pp. 861–874 (cit. on pp. 14, 17, 75).

[Felzenszwalb, 2004] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. "Efficient Graph-Based Image Segmentation". *International Journal of Computer Vision* 59.2 (2004), pp. 167–181 (cit. on p. 53).

[Ferri, 1994a] F. J. Ferri, P. Pudil, M. Hatef, and J. Kittler. "Comparative study of techniques for large-scale feature selection". *Pattern Recognition in Practice IV*. Ed. by Edzard S. GELSEMA and Laveen S. KANAL. Vol. 16. Machine Intelligence and Pattern Recognition. North-Holland, 1994, pp. 403–413 (cit. on p. 71).

[Ferri, 1994b] F.J. Ferri, P. Pudil, M. Hatef, and J. Kittler. "Comparative study of techniques for large-scale feature selection". *Pattern Recognition in Practice IV*. Ed. by Edzard S. GELSEMA and Laveen S. KANAL. Vol. 16. Machine Intelligence and Pattern Recognition. North-Holland, 1994, pp. 403–413 (cit. on pp. 94, 95).

[Galesso, 2023] Silvio Galesso, Max Argus, and Thomas Brox. "Far Away in the Deep Space: Dense Nearest-Neighbor-Based Out-of-Distribution Detection". Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023, pp. 4477–4487 (cit. on p. 8).

[Gangopadhyay, 2022] Tryambak Gangopadhyay, Sungmin Hong, Sujoy Roy, Yash Shah, and Lin Lee Cheong. "Rethinking benchmarking framework of self-supervised learning approaches for anomaly localization". *NeurIPS 2022 Workshop on Self-Supervised Learning - Theory and Practice*. 2022 (cit. on p. 20).

[Genc, 2021] Ergin U. Genc, Nilesh Ahuja, Ibrahima J. Ndiour, and Omesh Tickoo. "Energy-Based Anomaly Detection and Localization". Energy Based Models Workshop - ICLR 2021. 2021 (cit. on p. 8).

[GermainCars, 2024] GermainCars. *Most Popular Car Colors | Study by iSeeCars*. 2024. URL: https://www.germaincars.com/most-popular-car-colors/ (cit. on p. 4).

[Gong, 2019] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, et al. "Memorizing Normality to Detect Anomaly: Memory-Augmented Deep Autoencoder for Unsupervised Anomaly Detection". *Int. Conf. Comput. Vis.* 2019, pp. 1705–1714 (cit. on p. 10).

[Goodfellow, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, et al. "Generative adversarial nets". *Advances in neural information processing systems* 27 (2014) (cit. on p. 11).

[Goyal, 2020] Sachin Goyal, Aditi Raghunathan, Moksh Jain, Harsha Vardhan Simhadri, and Prateek Jain. *DROCC: Deep Robust One-Class Classification*. 2020 (cit. on p. 9).

[Gudovskiy, 2022] Denis Gudovskiy, Shun Ishizaka, and Kazuki Kozuka. "CFLOW-AD: Real-Time Unsupervised Anomaly Detection With Localization via Conditional Normalizing Flows". *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2022, pp. 98–107 (cit. on p. 9).

[Gui, 2022] Xingtai Gui, Di Wu, Yang Chang, and Shicai Fan. "Constrained Adaptive Projection with Pretrained Features for Anomaly Detection". Thirty-First International Joint Conference on Artificial Intelligence. Vol. 3. 2022, pp. 2059–2065 (cit. on p. 10).

[Gula, 2023] Tetiana Gula and João P. C. Bertoldo. "Gaussian Image Anomaly Detection with Greedy Eigencomponent Selection". *Int. Conf. Comput. Vis.* 2023, pp. 4110–4118 (cit. on pp. 7, 44, 75, 76, 79, 80).

[Han, 2022] Songqiao Han, Xiyang Hu, Hailiang Huang, Minqi Jiang, and Yue Zhao. "ADBench: Anomaly Detection Benchmark". Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track. 2022 (cit. on p. 6).

[He, 2015] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. *Deep Residual Learning for Image Recognition*. 2015 (cit. on p. 101).

[He, 2003] Zengyou He, Xiaofei Xu, and Shengchun Deng. "Discovering cluster-based local outliers". *Pattern Recognition Letters* 24.9 (2003), pp. 1641–1650 (cit. on p. 3).

[Hendrycks, 2022] Dan Hendrycks, Steven Basart, Mantas Mazeika, Andy Zou, Joseph Kwon, Mohammadreza Mostajabi, et al. "Scaling Out-of-Distribution Detection for Real-World Settings". *Int. Conf. Mach. Learn.* International Conference on Machine Learning. PMLR, 2022, pp. 8759–8773 (cit. on p. 5).

[Hojjati, 2024] Hadi Hojjati, Thi Kieu Khanh Ho, and Narges Armanfard. "Self-supervised anomaly detection in computer vision and beyond: A survey and outlook". *Neural Networks* 172 (2024), p. 106106 (cit. on p. 6).

[Huang, 2019] Weibo Huang and Peng Wei. *A PCB Dataset for Defects Detection and Classification*. 2019. arXiv: 1901.08204 (cit. on p. 5).

[Huang, 2018] Yibin Huang, Congying Qiu, Yue Guo, Xiaonan Wang, and Kui Yuan. "Surface Defect Saliency of Magnetic Tile". *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*. 2018 IEEE 14th International Conference on Automation Science and Engineering (CASE). 2018, pp. 612–617 (cit. on p. 5).

[Imamura, 2021] Ryuji Imamura, Kohei Azuma, Atsushi Hanamoto, and Atsunori Kanemura. *MLF-SC: Incorporating multi-layer features to sparse coding for anomaly detection*. 2021 (cit. on p. 10).

[Jang, 2023] JunKyu Jang, Eugene Hwang, and Sung-Hyuk Park. "N-Pad: Neighboring Pixel-Based Industrial Anomaly Detection". Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023, pp. 4365–4374 (cit. on pp. 8, 65, 121).

[Jeong, 2023] Jongheon Jeong, Yang Zou, Taewan Kim, Dongqing Zhang, Avinash Ravichandran, and Onkar Dabeer. "WinCLIP: Zero-/Few-Shot Anomaly Classification and Segmentation". *IEEE Conf. Comput. Vis. Pattern Recog.* 2023, pp. 19606–19616 (cit. on pp. 11, 20, 51).

[Jezek, 2022] Stepan Jezek, Martin Jonak, Radim Burget, Pavel Dvorak, and Milos Skotak. "Anomaly detection for real-world industrial applications: benchmarking recent self-supervised and pretrained methods". *2022 14th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*. 2022 14th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT). 2022, pp. 64–69 (cit. on p. 5).

[Jezequel, 2022] Loic Jezequel, Ngoc-Son Vu, Jean Beaudet, and Aymeric Histace. *Anomaly Detection via Multi-Scale Contrasted Memory*. 2022 (cit. on p. 10).

[Jiang, 2023] Zeyu Jiang, João P. C. Bertoldo, and Etienne Decencière. "Heuristic Hyperparameter Choice for Image Anomaly Detection". *2023 Twelfth International Conference on Image Processing Theory, Tools and Applications (IPTA)*. 2023, pp. 1–5 (cit. on p. 7).

[Jiang, 2024] Zhiqian Jiang, Yu Zhang, Yong Wang, Jinlong Li, and Xiaorong Gao. "FR-PatchCore: An Industrial Anomaly Detection Method for Improving Generalization". *Sensors* 24.5 (2024), p. 1368 (cit. on pp. 8, 44).

[Jonak, 2022] Martin Jonak, Stepan Jezek, and Radim Burget. "Evaluation of Nested U-Net models performance on MVTec AD dataset". *2022 14th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*. 2022 14th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT). 2022, pp. 70–75 (cit. on p. 10).

[Jung, 2022] Junyong Jung, Seungoh Han, Jinsun Park, and Donghyeon Cho. "A Comprehensive Real-World Photometric Stereo Dataset for Unsupervised Anomaly Detection". *IEEE Access* 10 (2022), pp. 108914–108923 (cit. on p. 5).

[Kamoi, 2020] Ryo Kamoi and Kei Kobayashi. *Why is the Mahalanobis Distance Effective for Anomaly Detection?* 2020 (cit. on p. 65).

[Kamoona, 2024] Ammar Mansoor Kamoona, Amirali Khodadadian Gostar, Xiaoying Wang, Mark Easton, Alireza Bab-Hadiashar, and Reza Hoseinnezhad. "Anomaly detection of defect using energy of point pattern features within random finite set framework". *Engineering Applications of Artificial Intelligence* 130 (2024), p. 107706 (cit. on p. 7).

[Kim, 2022a] Donghyeong Kim, Chaewon Park, Suhwan Cho, and Sangyoun Lee. *FAPM: Fast Adaptive Patch Memory for Real-time Industrial Anomaly Detection*. 2022 (cit. on p. 8).

[Kim, 2021] Jin-Hwa Kim, Do-Hyeong Kim, Saehoon Yi, and Taehoon Lee. *Semi-orthogonal Embedding for Efficient Unsupervised Anomaly Segmentation*. 2021 (cit. on pp. 7, 8, 65, 68, 74, 103, 121).

[Kim, 2022b] Yeongmin Kim, Huiwon Jang, DongKeon Lee, and Ho-Jin Choi. *AltUB: Alternating Training Method to Update Base Distribution of Normalizing Flow for Anomaly Detection*. 2022 (cit. on p. 9).

[Krizhevsky, 2009] Alex Krizhevsky. *Learning multiple layers of features from tiny images*. 2009 (cit. on p. 5).

[Krohling, 2019] 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 (cit. on p. 14).

[Kroon, 2009] Dirk-Jan Kroon. "Numerical Optimization of Kernel-Based Image Derivatives". *Short Paper University Twente* (2009) (cit. on p. 53).

[Lafon, 2023] Marc Lafon, Elias Ramzi, Clément Rambour, and Nicolas Thome. "Hybrid Energy Based Model in the Feature Space for Out-of-Distribution Detection" (2023) (cit. on p. 8).

[Lam, 2015] Siu Kwan Lam, Antoine Pitrou, and Stanley Seibert. "Numba: a LLVM-based Python JIT compiler". *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*. 2015, pp. 1–6 (cit. on p. 26).

[Lecun, 1998] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition". *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324 (cit. on p. 5).

[Lecun, 2006] Yann Lecun, Sumit Chopra, Raia Hadsell, Marc Aurelio Ranzato, and Fu Jie Huang. "A tutorial on energy-based learning". *Predicting structured data*. Ed. by G. Bakir, T. Hofman, B. Scholkopt, A. Smola, and B. Taskar. MIT Press, 2006 (cit. on p. 8).

[Ledoit, 2003] Olivier Ledoit and Michael Wolf. *Honey, I Shrunk the Sample Covariance Matrix*. Rochester, NY, 2003 (cit. on p. 68).

[Ledoit, 2004] Olivier Ledoit and Michael Wolf. "A well-conditioned estimator for large-dimensional covariance matrices". *Journal of Multivariate Analysis* 88.2 (2004), pp. 365–411 (cit. on p. 68).

[Lee, 2018] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. "A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks". *Adv. Neural Inform. Process. Syst.* Vol. 31. 2018 (cit. on p. 65).

[Lee, 2022] Sungwook Lee, Seunghyun Lee, and Byung Cheol Song. *CFA: Coupled-hypersphere-based Feature Adaptation for Target-Oriented Anomaly Localization*. 2022 (cit. on p. 9).

[Lei, 2023] Jiarui Lei, Xiaobo Hu, Yue Wang, and Dong Liu. "PyramidFlow: High-Resolution Defect Contrastive Localization Using Pyramid Normalizing Flow". *IEEE Conf. Comput. Vis. Pattern Recog.* 2023, pp. 14143–14152 (cit. on pp. 9, 31).

[Leys, 2018] Christophe Leys, Olivier Klein, Yves Dominicy, and Christophe Ley. "Detecting multivariate outliers: Use a robust variant of the Mahalanobis distance". *Journal of Experimental Social Psychology* 74 (2018), pp. 150–156 (cit. on pp. 3, 7).

[Li, 2021a] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. "CutPaste: Self-Supervised Learning for Anomaly Detection and Localization". *IEEE Conf. Comput. Vis. Pattern Recog.* 2021, pp. 9664–9674 (cit. on p. 11).

[Li, 2021b] Ning Li, Kaitao Jiang, Zhiheng Ma, Xing Wei, Xiaopeng Hong, and Yihong Gong. "Anomaly Detection Via Self-Organizing Map". *IEEE Int. Conf. Image Process.* 2021, pp. 974–978 (cit. on p. 8).

[Li, 2020] Zhenyu Li, Ning Li, Kaitao Jiang, Zhiheng Ma, Xing Wei, Xiaopeng Hong, et al. "Superpixel Masking and Inpainting for Self-Supervised Anomaly Detection". BMVC. 2020, p. 12 (cit. on p. 11).

[Lin, 2022] Jie Lin, Song Chen, Enping Lin, and Yu Yang. "Deep Feature Selection for Anomaly Detection Based on Pretrained Network and Gaussian Discriminative Analysis". *IEEE Open Journal of Instrumentation and Measurement* 1 (2022), pp. 1–11 (cit. on pp. 7, 65, 68, 73, 76–78, 83, 103, 121, 173).

[Lis, 2019] Krzysztof Lis, Krishna Nakka, Pascal Fua, and Mathieu Salzmann. "Detecting the Unexpected via Image Resynthesis". *Int. Conf. Comput. Vis.* 2019, pp. 2152–2161 (cit. on pp. 5, 10).

[Liu, 2020] Fenglin Liu, Xuancheng Ren, Zhiyuan Zhang, Xu Sun, and Yuexian Zou. "Rethinking Skip Connection with Layer Normalization". *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, 2020, pp. 3586–3598 (cit. on p. 76).

[Liu, 2018] W. Liu, D. Lian W. Luo, and S. Gao. "Future Frame Prediction for Anomaly Detection – A New Baseline". *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018 (cit. on p. 5).

[Liu, 2021] Yunfei Liu, Chaoqun Zhuang, and Feng Lu. *Unsupervised Two-Stage Anomaly Detection*. 2021 (cit. on p. 10).

[Liu, 2023] Zhikang Liu, Yiming Zhou, Yuansheng Xu, and Zilei Wang. "SimpleNet: A Simple Network for Image Anomaly Detection and Localization". *IEEE Conf. Comput. Vis. Pattern Recog.* 2023, pp. 20402–20411 (cit. on pp. 11, 31, 97).

[Liznerski, 2021] Philipp Liznerski, Lukas Ruff, Robert A. Vandermeulen, Billy Joe Franks, Marius Kloft, and Klaus Robert Muller. "Explainable Deep One-Class Classification". *Int. Conf. Learn. Represent.* 2021 (cit. on pp. 9, 12, 122, 123, 125–129).

[Liznerski, 2022] Philipp Liznerski, Lukas Ruff, Robert A. Vandermeulen, Billy Joe Franks, Klaus Robert Muller, and Marius Kloft. "Exposing Outlier Exposure: What Can Be Learned From Few, One, and Zero Outlier Images". *Transactions on Machine Learning Research* (2022) (cit. on p. 9).

[Lu, 2013] Cewu Lu, Jianping Shi, and Jiaya Jia. "Abnormal Event Detection at 150 FPS in MATLAB". Proceedings of the IEEE International Conference on Computer Vision. 2013, pp. 2720–2727 (cit. on p. 5).

[Luo, 2017] Weixin Luo, Wen Liu, and Shenghua Gao. "A Revisit of Sparse Coding Based Anomaly Detection in Stacked RNN Framework". Proceedings of the IEEE International Conference on Computer Vision. 2017, pp. 341–349 (cit. on p. 5).

[Madan, 2022] Neelu Madan, Nicolae-Catalin Ristea, Radu Tudor Ionescu, Kamal Nasrollahi, Fahad Shahbaz Khan, Thomas B. Moeslund, et al. *Self-Supervised Masked Convolutional Transformer Block for Anomaly Detection*. 2022 (cit. on p. 10).

[Mahadevan, 2010] Vijay Mahadevan, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos. "Anomaly detection in crowded scenes". *IEEE Conf. Comput. Vis. Pattern Recog.* 2010, pp. 1975–1981 (cit. on pp. 5, 14, 39).

[Mahalanobis, 1936] P. C. Mahalanobis. "On the generalized distance in Statistics" (1936) (cit. on p. 67).

[Massoli, 2022] Fabio Valerio Massoli, Fabrizio Falchi, Alperen Kantarci, Şeymanur Akti, Hazim Kemal Ekenel, and Giuseppe Amato. "MOCCA: Multi-Layer One-Class ClassificAtion for Anomaly Detection". *IEEE Transactions on Neural Networks and Learning Systems* 33.6 (2022), pp. 2313–2323 (cit. on p. 9).

[Mirzaei, 2022] Hossein Mirzaei, Mohammadreza Salehi, Sajjad Shahabi, Efstratios Gavves, Cees G. M. Snoek, Mohammad Sabokrou, et al. *Fake It Till You Make It: Near-Distribution Novelty Detection by Score-Based Generative Models*. arXiv:2205.14297. arXiv, 2022 (cit. on p. 11).

[Mishra, 2021] Pankaj Mishra, Riccardo Verk, Daniele Fornasier, Claudio Piciarelli, and Gian Luca Foresti. "VT-ADL: A Vision Transformer Network for Image Anomaly Detection and Localization". *2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)*. 2021, pp. 01–06 (cit. on pp. 5, 14).

[Ndiour, 2022] Ibrahima Ndiour, Nilesh Ahuja, Utku Genc, and Omesh Tickoo. *FRE: A Fast Method For Anomaly Detection And Segmentation*. 2022. arXiv: 2211.12650 (cit. on p. 10).

[Neubert, 2014] Peer Neubert and Peter Protzel. "Compact Watershed and Preemptive SLIC: On Improving Trade-offs of Superpixel Segmentation Algorithms". *Int. Conf. Pattern Recog.* 2014, pp. 996–1001 (cit. on pp. 53, 60).

[Pang, 2021a] Guansong Pang and Charu Aggarwal. "Toward Explainable Deep Anomaly Detection". *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. Virtual Event Singapore: ACM, 2021, pp. 4056–4057 (cit. on p. 6).

[Pang, 2021b] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. "Deep Learning for Anomaly Detection: A Review". *ACM Comput. Surv.* 54.2 (2021) (cit. on p. 6).

[Papamakarios, 2021] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. "Normalizing Flows for Probabilistic Modeling and Inference". *J. Mach. Learn. Res.* 22.57 (2021), pp. 1–64 (cit. on p. 9).

[Park, 2020] Hyunjong Park, Jongyoun Noh, and Bumsub Ham. "Learning Memory-Guided Normality for Anomaly Detection". *IEEE Conf. Comput. Vis. Pattern Recog.* Seattle, WA, USA: IEEE, 2020, pp. 14360–14369 (cit. on p. 10).

[Perera, 2021] Pramuditha Perera, Poojan Oza, and Vishal M. Patel. *One-Class Classification: A Survey*. 2021 (cit. on pp. 9, 123).

[Pranav, 2020] Mantini Pranav, Li Zhenggang, and Shah Shishir K. "A Day on Campus - An Anomaly Detection Dataset for Events in a Single Camera". *Asian Conf. Comput. Vis.* 2020 (cit. on pp. 5, 14, 39).

[Radford, 2021] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, et al. "Learning Transferable Visual Models From Natural Language Supervision". *Int. Conf. Mach. Learn.* Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 8748–8763 (cit. on p. 11).

[Radford, 2016] Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks". *Int. Conf. Learn. Represent.* Ed. by Yoshua Bengio and Yann LeCun. 2016 (cit. on p. 11).

[Rafiei, 2023] Mehdi Rafiei, Toby P. Breckon, and Alexandros Iosifidis. *On Pixel-level Performance Assessment in Anomaly Detection.* 2023 (cit. on pp. 14, 17, 20).

[Raghavendra, 2019] Chalapathy Raghavendra. "Deep Learning for Anomaly Detection". PhD thesis. 2019 (cit. on p. 6).

[Ramachandra, 2020] Bharathkumar Ramachandra and Michael J. Jones. "Street Scene: A new dataset and evaluation protocol for video anomaly detection". *2020 IEEE Winter Conference on Applications of Computer Vision (WACV).* 2020 IEEE Winter Conference on Applications of Computer Vision (WACV). Snowmass Village, CO, USA: IEEE, 2020, pp. 2558–2567 (cit. on pp. 5, 14, 39).

[Reiss, 2021] Tal Reiss, Niv Cohen, Liron Bergman, and Yedid Hoshen. "PANDA: Adapting Pretrained Features for Anomaly Detection and Segmentation". *IEEE Conf. Comput. Vis. Pattern Recog.* 2021, pp. 2806–2814 (cit. on p. 11).

[Riba, 2020] Edgar Riba, Dmytro Mishkin, Daniel Ponsa, Ethan Rublee, and Gary Bradski. "Kornia: an Open Source Differentiable Computer Vision Library for PyTorch". *IEEE/CVF Winter Conf. Appl. Comput. Vis.* 2020, pp. 3674–3683 (cit. on p. 25).

[Rippel, 2021a] O. Rippel, P. Mertens, and D. Merhof. "Modeling the Distribution of Normal Data in Pre-Trained Deep Features for Anomaly Detection". *Int. Conf. Pattern Recog.* 2021, pp. 6726–6733 (cit. on pp. 7, 65, 68, 78, 93, 103, 121).

[Rippel, 2021b] Oliver Rippel and Dorit Merhof. "Leveraging pre-trained Segmentation Networks for Anomaly Segmentation". *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA).* 2021, pp. 01–04 (cit. on p. 7).

[Rippel, 2021c] Oliver Rippel, Patrick Mertens, Eike König, and Dorit Merhof. "Gaussian Anomaly Detection by Modeling the Distribution of Normal Data in Pretrained Deep Features". *IEEE Transactions on Instrumentation and Measurement* 70 (2021), pp. 1–13 (cit. on pp. 7, 44, 65, 68, 76, 103, 121).

[Rippel, 2022] Oliver Rippel., Arnav Chavan., Chucai Lei., and Dorit Merhof. "Transfer Learning Gaussian Anomaly Detection by Fine-tuning Representations". *Proceedings of the 2nd International Conference on Image Processing and Vision Engineering - IMPROVE.* INSTICC. SciTePress, 2022, pp. 45–56 (cit. on p. 8).

[Ronneberger, 2015a] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015.* Ed. by Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi. Cham: Springer International Publishing, 2015, pp. 234–241 (cit. on p. 9).

[Ronneberger, 2015b] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". *Medical Image Computing and Computer-Assisted Intervention.* 2015, pp. 234–241 (cit. on p. 128).

[Roth, 2022] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. "Towards Total Recall in Industrial Anomaly Detection". *IEEE Conf. Comput. Vis. Pattern Recog.* 2022, pp. 14318–14328 (cit. on pp. 8, 31, 39, 44, 60, 74, 78, 97).

[Rudolph, 2021] Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn. "Same Same but DifferNet: Semi-Supervised Defect Detection With Normalizing Flows". Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2021, pp. 1907–1916 (cit. on p. 9).

[Rudolph, 2022] Marco Rudolph, Tom Wehrbein, Bodo Rosenhahn, and Bastian Wandt. "Fully Convolutional Cross-Scale-Flows for Image-Based Defect Detection". Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2022, pp. 1088–1097 (cit. on p. 9).

[Ruff, 2021a] Lukas Ruff, Jacob R. Kauffmann, Robert A. Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, et al. "A Unifying Review of Deep and Shallow Anomaly Detection". *Proceedings of the IEEE* 109.5 (2021), pp. 756–795 (cit. on pp. 3, 6).

[Ruff, 2018] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, et al. "Deep One-Class Classification". *Int. Conf. Mach. Learn.* Ed. by Jennifer Dy and Andreas Krause. Vol. 80. PMLR, 2018, pp. 4393–4402 (cit. on pp. 9, 123).

[Ruff, 2021b] Lukas Ruff, Robert A. Vandermeulen, Billy Joe Franks, Klaus-Robert Müller, and Marius Kloft. *Rethinking Assumptions in Deep Anomaly Detection*. 2021 (cit. on p. 123).

[Ruff, 2020] Lukas Ruff, Robert A. Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, et al. "Deep Semi-Supervised Anomaly Detection". *Int. Conf. Learn. Represent.* 2020 (cit. on pp. 9, 123).

[Saito, 2015] 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 (2015), e0118432 (cit. on pp. 14, 17).

[Sakurada, 2014] Mayu Sakurada and Takehisa Yairi. "Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction". *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*. MLSDA'14. New York, NY, USA: Association for Computing Machinery, 2014, pp. 4–11 (cit. on p. 10).

[Salehi, 2021a] Mohammadreza Salehi, Atrin Arya, Barbod Pajoum, Mohammad Otoofi, Amirreza Shaeiri, Mohammad Hossein Rohban, et al. "ARAE: Adversarially robust training of autoencoders improves novelty detection". *Neural Networks* 144 (2021), pp. 726–736 (cit. on p. 10).

[Salehi, 2022] Mohammadreza Salehi, Hossein Mirzaei, Dan Hendrycks, Yixuan Li, Mohammad Hossein Rohban, and Mohammad Sabokrou. "A Unified Survey on Anomaly, Novelty, Open-Set, and Out of-Distribution Detection: Solutions and Future Challenges". *Transactions on Machine Learning Research* (2022) (cit. on p. 3).

[Salehi, 2021b] Mohammadreza Salehi, Niousha Sadjadi, Soroosh Baselizadeh, Mohammad H. Rohban, and Hamid R. Rabiee. "Multiresolution Knowledge Distillation for Anomaly Detection". *IEEE Conf. Comput. Vis. Pattern Recog.* 2021, pp. 14902–14912 (cit. on p. 11).

[Samele, 2022] Stefano Samele and Matteo Matteucci. "Patchwise Sparse Dictionary Learning from pre-trained Neural Network Activation Maps for Anomaly Detection in Images". *Int. Conf. Pattern Recog.* 2022, pp. 1307–1313 (cit. on p. 10).

[Schlegl, 2019] Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Georg Langs, and Ursula Schmidt-Erfurth. "f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks". *Medical Image Analysis* 54 (2019), pp. 30–44 (cit. on p. 11).

[Schlegl, 2017] Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. "Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery". *Information Processing in Medical Imaging*. Ed. by Marc Niethammer, Martin Styner, Stephen Aylward, Hongtu Zhu, Ipek Oguz, Pew-Thian Yap, et al. Cham: Springer International Publishing, 2017, pp. 146–157 (cit. on p. 11).

[Schlüter, 2022] Hannah M. Schlüter, Jeremy Tan, Benjamin Hou, and Bernhard Kainz. "Natural Synthetic Anomalies for Self-supervised Anomaly Detection and Localization". *Eur. Conf. Comput. Vis.* Ed. by Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner. Cham: Springer Nature Switzerland, 2022, pp. 474–489 (cit. on p. 11).

[Schölkopf, 2001] Bernhard Schölkopf, John C. Platt, John Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. "Estimating the Support of a High-Dimensional Distribution". *Neural Computation* 13.7 (2001), pp. 1443–1471 (cit. on pp. 9, 123).

[Schwartz, 2022] Eli Schwartz, Assaf Arbelle, Leonid Karlinsky, Sivan Harary, Florian Scheidegger, Sivan Doveh, et al. *MAEDAY: MAE for few and zero shot AnomalY-Detection*. 2022 (cit. on p. 10).

[Shi, 2021] Yong Shi, Jie Yang, and Zhiquan Qi. "Unsupervised anomaly segmentation via deep feature reconstruction". *Neurocomputing* 424 (2021), pp. 9–22 (cit. on p. 10).

[Sohn, 2021] Kihyuk Sohn, Chun-Liang Li, Jinsung Yoon, Minho Jin, and Tomas Pfister. "Learning and Evaluating Representations for Deep One-Class Classification". *International Conference on Learning Representations*. 2021 (cit. on p. 11).

[Su, 2023] Binyi Su, Zhong Zhou, and Haiyong Chen. "PVEL-AD: A Large-Scale Open-World Dataset for Photovoltaic Cell Anomaly Detection". *IEEE Transactions on Industrial Informatics* 19.1 (2023), pp. 404–413 (cit. on p. 5).

[Tabernik, 2020] Domen Tabernik, Samo Šela, Jure Skvarč, and Danijel Skočaj. "Segmentation-based deep-learning approach for surface-defect detection". *Journal of Intelligent Manufacturing* 31.3 (2020), pp. 759–776 (cit. on p. 5).

[Tack, 2020] Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. "CSI: Novelty Detection via Contrastive Learning on Distributionally Shifted Instances". *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 11839–11852 (cit. on p. 11).

[Tailanian, 2021] Matías Tailanian, Pablo Musé, and Álvaro Pardo. "A Multi-Scale A Contrario method for Unsupervised Image Anomaly Detection". *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA). 2021, pp. 179–184 (cit. on pp. 44, 45).

[Tailanian, 2024] Matías Tailanian, Álvaro Pardo, and Pablo Musé. "U-Flow: A U-Shaped Normalizing Flow for Anomaly Detection with Unsupervised Threshold". *Journal of Mathematical Imaging and Vision* 66.4 (2024), pp. 678–696 (cit. on pp. 9, 31, 44, 45, 60).

[Tan, 2019] Mingxing Tan and Quoc Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 2019, pp. 6105–6114 (cit. on pp. 76, 112).

[Tao, 2022] Xian Tao, Xinyi Gong, Xin Zhang, Shaohua Yan, and Chandranath Adak. "Deep Learning for Unsupervised Anomaly Localization in Industrial Images: A Survey". *IEEE Transactions on Instrumentation and Measurement* 71 (2022), pp. 1–21 (cit. on p. 6).

[Tax, 2004] David M.J. Tax and Robert P.W. Duin. "Support Vector Data Description". *Machine Learning* 54.1 (2004), pp. 45–66 (cit. on pp. 9, 123).

[Tien, 2023] Tran Dinh Tien, Anh Tuan Nguyen, Nguyen Hoang Tran, Ta Duc Huy, Soan T. M. Duong, Chanh D. Tr Nguyen, et al. "Revisiting Reverse Distillation for Anomaly Detection". *IEEE Conf. Comput. Vis. Pattern Recog.* 2023, pp. 24511–24520 (cit. on pp. 11, 31, 44, 60, 97).

[Tsai, 2022] Chin-Chia Tsai, Tsung-Hsuan Wu, and Shang-Hong Lai. "Multi-Scale Patch-Based Representation Learning for Image Anomaly Detection and Segmentation". *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). 2022, pp. 3065–3073 (cit. on p. 11).

[Tschuchnig, 2022] Maximilian E. Tschuchnig, Philipp Grubmüller, Lea M. Stangassinger, Christina Kreutzer, Sebastien Couillard-Després, Gertie J. Oostingh, et al. "Evaluation of Multi-Scale Multiple Instance Learning to Improve Thyroid Cancer Classification". *2022 Eleventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*. 2022, pp. 1–6 (cit. on p. 7).

[Vedaldi, 2008] Andrea Vedaldi and Stefano Soatto. "Quick Shift and Kernel Methods for Mode Seeking". *Eur. Conf. Comput. Vis.* 2008, pp. 705–718 (cit. on p. 53).

[Venkataramanan, 2020] Shashanka Venkataramanan, Kuan-Chuan Peng, Rajat Vikram Singh, and Abhijit Mahalanobis. "Attention Guided Anomaly Localization in Images". *Eur. Conf. Comput. Vis.* Ed. by Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm. Cham: Springer International Publishing, 2020, pp. 485–503 (cit. on p. 10).

[Virtanen, 2020] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". *Nature Methods* 17 (2020), pp. 261–272 (cit. on p. 56).

[Walt, 2014] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, et al. "scikit-image: image processing in Python". *PeerJ* 2 (2014), e453 (cit. on p. 60).

[Wan, 2022a] Qian Wan, Liang Gao, Xinyu Li, and Long Wen. "Industrial Image Anomaly Localization Based on Gaussian Clustering of Pretrained Feature". *IEEE Transactions on Industrial Electronics* 69.6 (2022), pp. 6182–6192 (cit. on pp. 8, 11).

[Wan, 2022b] Qian Wan, Liang Gao, Xinyu Li, and Long Wen. "Unsupervised Image Anomaly Detection and Segmentation Based on Pre-Trained Feature Mapping". *IEEE Transactions on Industrial Informatics* (2022), pp. 1–10 (cit. on p. 11).

[Wang, 2020a] Lu Wang, Dongkai Zhang, Jiahao Guo, and Yuexing Han. "Image Anomaly Detection Using Normal Data Only by Latent Space Resampling". *Applied Sciences* 10.23 (2020), p. 8660 (cit. on p. 10).

[Wang, 2023] Pei Wang, Wei Zhai, Yang Cao, University of Science and Technology of China, and Institute of Artificial Intelligence, Hefei Comprehensive National Science Center. "Robustness Benchmark for Unsupervised Anomaly Detection Models". *JUSTC* 53.0 (2023), p. 1 (cit. on pp. 3, 5).

[Wang, 2020b] Yaqing Wang, Quanming Yao, James Kwok, and Lionel M. Ni. *Generalizing from a Few Examples: A Survey on Few-Shot Learning*. 2020 (cit. on p. 6).

[Wei, 2023] Shenxing Wei, Xing Wei, Muhammad Rifki Kurniawan, Zhiheng Ma, and Yihong Gong. "Topology-preserving transfer learning for weakly-supervised anomaly detection and segmentation". *Pattern Recognition Letters* 170 (2023), pp. 77–84 (cit. on p. 11).

[Xiao, 2017] Han Xiao, Kashif Rasul, and Roland Vollgraf. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. 2017. arXiv: 1708.07747 (cit. on p. 5).

[Xu, 2023] Hongzuo Xu, Guansong Pang, Yijie Wang, and Yongjun Wang. "Deep Isolation Forest for Anomaly Detection". *IEEE Transactions on Knowledge and Data Engineering* 35.12 (2023), pp. 12591–12604 (cit. on p. 3).

[Yang, 2021] Jie Yang, Yong Shi, and Zhiquan Qi. "DFR: Deep Feature Reconstruction for Unsupervised Anomaly Segmentation". *Neurocomputing* 424 (2021), pp. 9–22 (cit. on p. 10).

[Yang, 2024] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. "Generalized Out-of-Distribution Detection: A Survey". *Int. J. Comput. Vis.* (2024) (cit. on p. 3).

[Yao, 2022] Haiming Yao and Xue Wang. *Generalizable Industrial Visual Anomaly Detection with Self-Induction Vision Transformer*. 2022 (cit. on p. 10).

[Yi, 2020] Jihun Yi and Sungroh Yoon. "Patch SVDD: Patch-level SVDD for Anomaly Detection and Segmentation". Proceedings of the Asian Conference on Computer Vision. 2020 (cit. on p. 8).

[Yu, 2020] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, et al. "BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning". Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020, pp. 2636–2645 (cit. on p. 5).

[Yu, 2021] Jiawei Yu, Ye Zheng, Xiang Wang, Wei Li, Yushuang Wu, Rui Zhao, et al. *FastFlow: Unsupervised Anomaly Detection and Localization via 2D Normalizing Flows*. 2021 (cit. on pp. 9, 31).

[Zavrtanik, 2021a] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. "DRAEM - A Discriminatively Trained Reconstruction Embedding for Surface Anomaly Detection". *Int. Conf. Comput. Vis.* 2021, pp. 8330–8339 (cit. on pp. 10, 89).

[Zavrtanik, 2021b] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. "Reconstruction by inpainting for visual anomaly detection". *Pattern Recognition* 112 (2021), p. 107706 (cit. on p. 10).

[Zhang, 2022a] Hui Zhang, Zuxuan Wu, Zheng Wang, Zhineng Chen, and Yu-Gang Jiang. *Prototypical Residual Networks for Anomaly Detection and Localization*. 2022 (cit. on p. 10).

[Zhang, 2023a] Jian Zhang, Runwei Ding, Miaoju Ban, and Ge Yang. *PKU-GoodsAD: A Supermarket Goods Dataset for Unsupervised Anomaly Detection and Segmentation*. 2023 (cit. on p. 5).

[Zhang, 2022b] Kaitai Zhang, Bin Wang, and C. -C. Jay Kuo. "PEDENet: Image anomaly localization via patch embedding and density estimation". *Pattern Recognition Letters* 153 (2022), pp. 144–150 (cit. on p. 8).

[Zhang, 2022c] Qianqian Zhang, Hongyang Wei, Xusheng Du, Xue Li, and Jiong Yu. "FAIAD: Feature Adaptive-based Image Anomaly Detection". *Journal of Physics: Conference Series* 2333.1 (2022), p. 012005 (cit. on p. 8).

[Zhang, 2022d] Xingbao Zhang, Wei Li, and Yue Zhao. "One-class Anomaly Detection with Redundancy Reduction and Momentum Mechanism". *2022 4th International Conference on Data-driven Optimization of Complex Systems (DOCS)*. 2022 4th International Conference on Data-driven Optimization of Complex Systems (DOCS). 2022, pp. 1–6 (cit. on p. 9).

[Zhang, 2023b] Xuan Zhang, Shiyu Li, Xi Li, Ping Huang, Jiulong Shan, and Ting Chen. "DeSTSeg: Segmentation Guided Denoising Student-Teacher for Anomaly Detection". *IEEE Conf. Comput. Vis. Pattern Recog.* 2023, pp. 3914–3923 (cit. on p. 11).

[Zhang, 2023c] Xuan Zhang, Shiyu Li, Xi Li, Ping Huang, Jiulong Shan, and Ting Chen. "DeSTSeg: Segmentation Guided Denoising Student-Teacher for Anomaly Detection". *IEEE Conf. Comput. Vis. Pattern Recog.* 2023, pp. 3914–3923 (cit. on pp. 17, 20, 32, 51).

[Zheng, 2022] Ye Zheng, Xiang Wang, Rui Deng, Tianpeng Bao, Rui Zhao, and Liwei Wu. "Focus Your Distribution: Coarse-to-Fine Non-Contrastive Learning for Anomaly Detection and Localization". *Int. Conf. Multimedia and Expo*. 2022, pp. 1–6 (cit. on pp. 8, 65, 121).

[Zhou, 2022a] Kang Zhou, Jing Li, Weixin Luo, Zhengxin Li, Jianlong Yang, Huazhu Fu, et al. "Proxy-Bridged Image Reconstruction Network for Anomaly Detection in Medical Images". *IEEE Transactions on Medical Imaging* 41.3 (2022), pp. 582–594 (cit. on p. 10).

[Zhou, 2022b] Kang Zhou, Jing Li, Yuting Xiao, Jianlong Yang, Jun Cheng, Wen Liu, et al. "Memorizing Structure-Texture Correspondence for Image Anomaly Detection". *IEEE Transactions on Neural Networks and Learning Systems* 33.6 (2022), pp. 2335–2349 (cit. on p. 10).

[Zhou, 2020] Kang Zhou, Yuting Xiao, Jianlong Yang, Jun Cheng, Wen Liu, Weixin Luo, et al. "Encoding Structure-Texture Relation with P-Net for Anomaly Detection in Retinal Images". *Eur. Conf. Comput. Vis.* Ed. by Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm. Cham: Springer International Publishing, 2020, pp. 360–377 (cit. on p. 10).

[Zhou, 2023] Qihang Zhou, Shibo He, Haoyu Liu, Tao Chen, and Jiming Chen. "Pull & Push: Leveraging Differential Knowledge Distillation for Efficient Unsupervised Anomaly Detection and Localization". *IEEE Transactions on Circuits and Systems for Video Technology* 33.5 (2023), pp. 2176–2189 (cit. on p. 11).

[Zimmerer, 2022] David Zimmerer, Peter M. Full, Fabian Isensee, Paul Jäger, Tim Adler, Jens Petersen, et al. "MOOD 2020: A Public Benchmark for Out-of-Distribution Detection and Localization on Medical Images". *IEEE Transactions on Medical Imaging* 41.10 (2022), pp. 2728–2738 (cit. on p. 5).

[Zimmerer, 2019] David Zimmerer, Fabian Isensee, Jens Petersen, Simon Kohl, and Klaus Maier-Hein. "Unsupervised Anomaly Localization Using Variational Auto-Encoders". *Medical Image Computing and Computer Assisted Intervention - MICCAI 2019*. Ed. by Dinggang Shen, Tianming Liu, Terry M. Peters, Lawrence H. Staib, Caroline Essert, Sean Zhou, et al. Cham: Springer International Publishing, 2019, pp. 289–297 (cit. on p. 10).

[Zingman, 2022] Igor Zingman, Birgit Stierstorfer, Charlotte Lempp, and Fabian Heinemann. *Learning image representations for anomaly detection: application to discovery of histological alterations in drug development.* 2022 (cit. on p. 5).

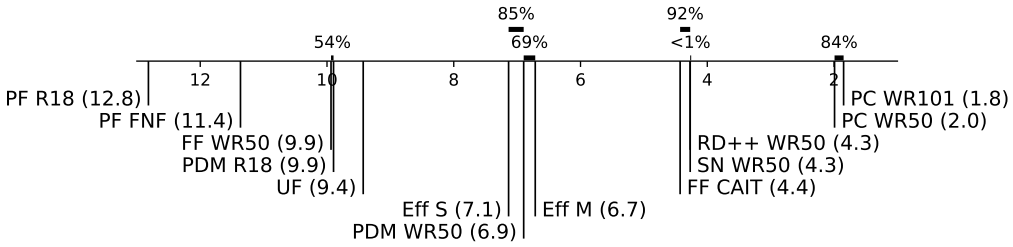
[Zou, 2022] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. "SPot-the-Difference Self-supervised Pre-training for Anomaly Detection and Segmentation". *Eur. Conf. Comput. Vis.* 2022, pp. 392–408 (cit. on pp. 5, 14, 20, 28, 51).

Appendix

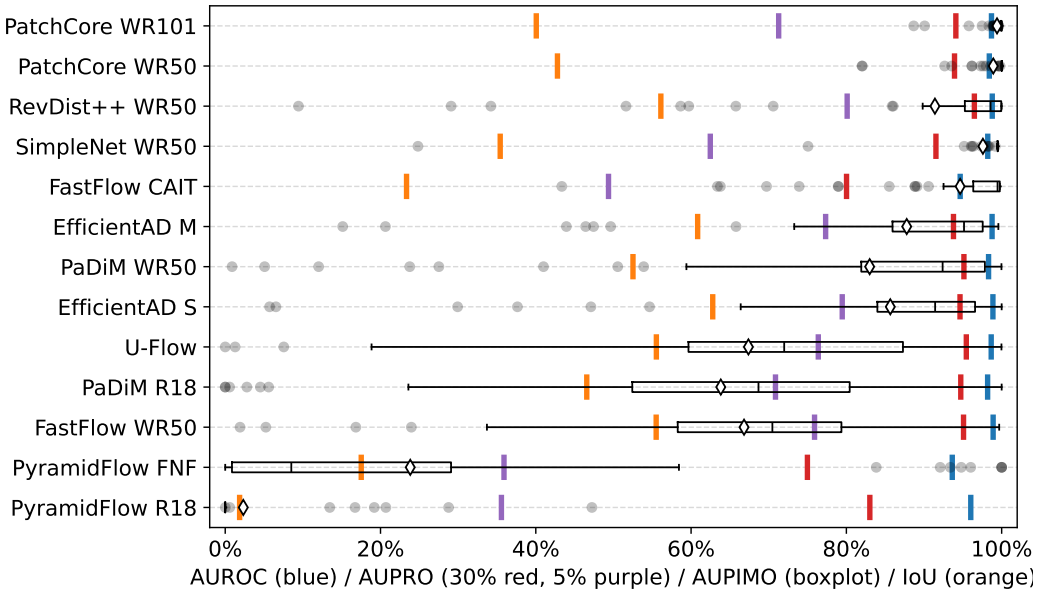
1 Benchmark

Model	Dataset Collection	AUROC	AUPRO	AUPIMO	Avg.	P_{33}	Avg. Rank
PaDiM R18	MVTec AD	96.62	91.58		25.75	14.34	10.5
PaDiM R18	VisA	97.22	81.87		16.42	4.33	10.1
PaDiM R18	All	96.89	87.27		21.61	9.89	10.3
FastFlow WR50	MVTec AD	97.01	90.87		28.49	14.15	10.3
FastFlow WR50	VisA	96.83	80.54		20.65	8.03	8.9
FastFlow WR50	All	96.93	86.28		25.00	11.43	9.7
PaDiM WR50	MVTec AD	97.19	92.57		40.14	27.06	8.9
PaDiM WR50	VisA	97.32	80.81		17.34	8.12	9.9
PaDiM WR50	All	97.25	87.35		30.01	18.64	9.3
PyramidFlow FNF	MVTec AD	94.21	79.10		36.26	19.94	9.4
PyramidFlow FNF	VisA	96.62	82.03		31.55	9.56	7.8
PyramidFlow FNF	All	95.28	80.40		34.17	15.33	8.7
PyramidFlow R18	MVTec AD	96.36	85.81		36.32	23.91	9.0
PyramidFlow R18	VisA	96.53	84.27		26.84	5.55	8.1
PyramidFlow R18	All	96.44	85.13		32.11	15.75	8.6
SimpleNet WR50	MVTec AD	97.13	89.48		71.39	62.78	5.3
SimpleNet WR50	VisA	91.17	69.88		34.66	17.93	7.4
SimpleNet WR50	All	94.48	80.77		55.07	42.84	6.3
PatchCore WR50	MVTec AD	98.01	93.13		67.21	54.95	5.6
PatchCore WR50	VisA	98.26	87.69		38.02	15.74	6.9
PatchCore WR50	All	98.12	90.72		54.24	37.53	6.1
EfficientAD S	MVTec AD	97.96	93.65		64.76	55.16	5.9
EfficientAD S	VisA	98.89	91.90		54.62	37.78	5.2
EfficientAD S	All	98.37	92.87		60.25	47.44	5.6
RevDist++ WR50	MVTec AD	98.23	95.03		71.93	64.93	4.9
RevDist++ WR50	VisA	99.00	91.53		44.30	15.85	6.3
RevDist++ WR50	All	98.57	93.48		59.65	43.11	5.6
FastFlow CAIT	MVTec AD	97.37	90.44		66.79	57.83	5.4
FastFlow CAIT	VisA	98.25	89.37		49.10	28.09	5.4
FastFlow CAIT	All	97.76	89.96		58.93	44.61	5.4
EfficientAD M	MVTec AD	97.96	94.10		66.08	55.97	5.8
EfficientAD M	VisA	99.00	92.25		58.06	40.52	4.6
EfficientAD M	All	98.42	93.28		62.52	49.10	5.2
U-Flow	MVTec AD	98.74	94.89		66.07	56.07	5.4
U-Flow	VisA	99.09	91.14		51.48	31.54	4.9
U-Flow	All	98.89	93.22		59.58	45.17	5.2
PatchCore WR101	MVTec AD	98.35	93.53		73.19	66.12	4.7
PatchCore WR101	VisA	98.70	90.06		48.72	31.58	5.5
PatchCore WR101	All	98.51	91.99		62.31	50.77	5.1

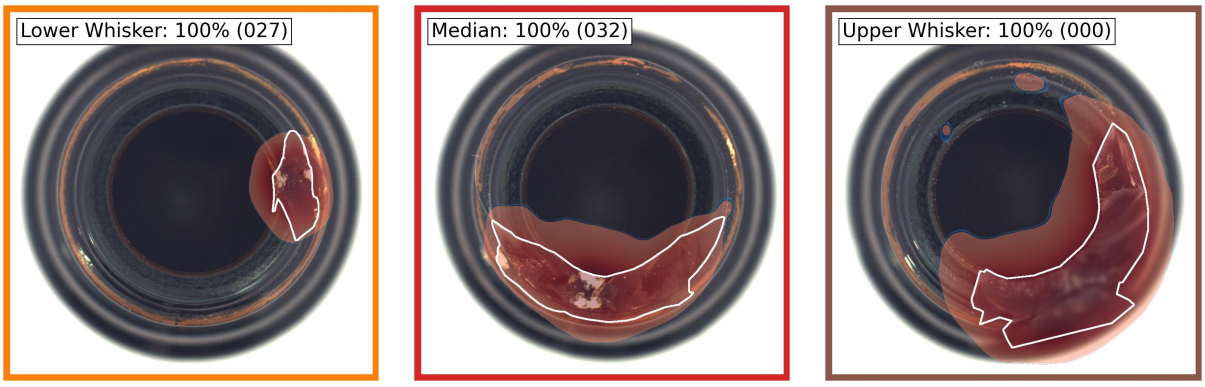
Table A.3: Model averages. Scores are in percentages. Ranks range from 1 (best) to number of models (worst).



(a) Average rank diagram.

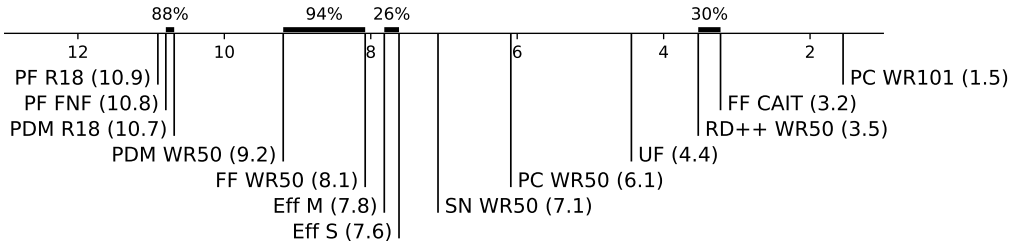


(b) Score distributions.

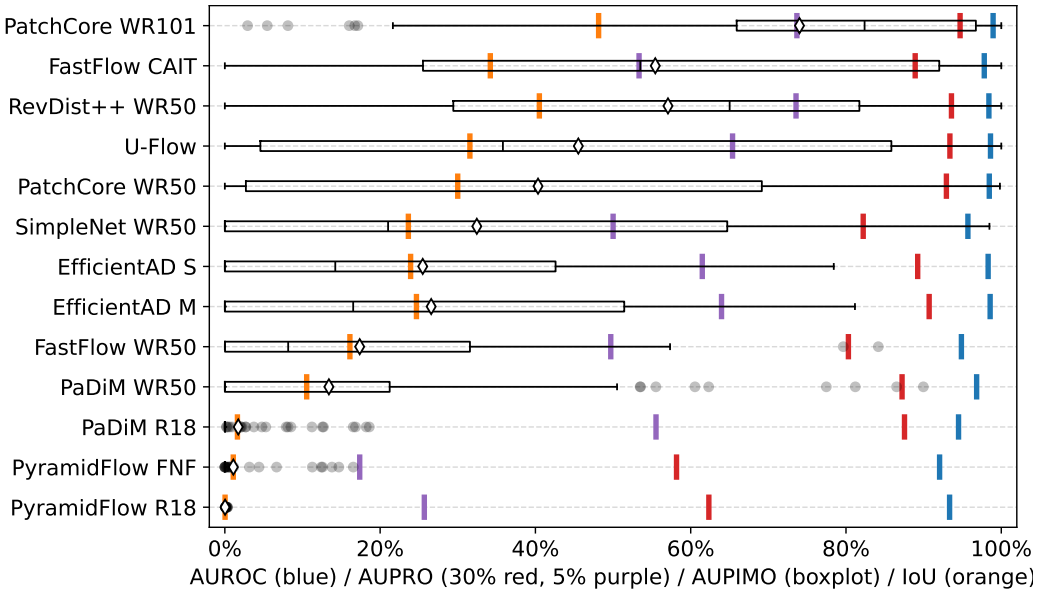


(c) Heatmaps. Images selected according to AUPIMO's statistics. Statistic and image index annotated on upper left corner.

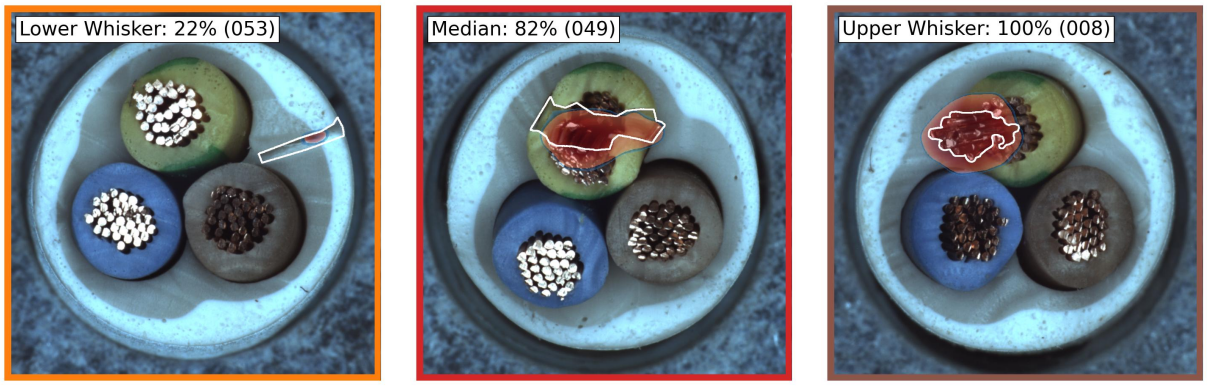
Figure A.4: Benchmark on MVTec AD / Bottle. PIMO curves and heatmaps are from PatchCore WR101. 083 images (020 normal, 063 anomalous).



(a) Average rank diagram.

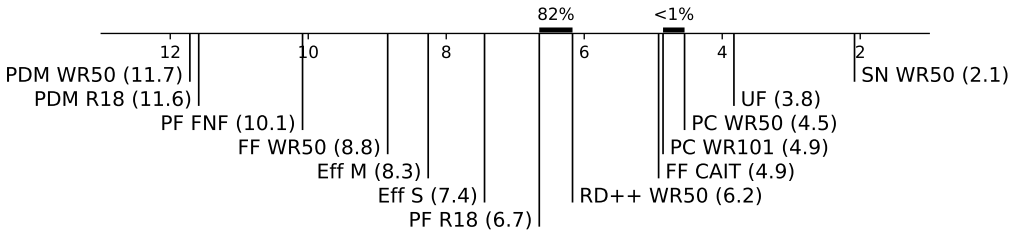


(b) Score distributions.

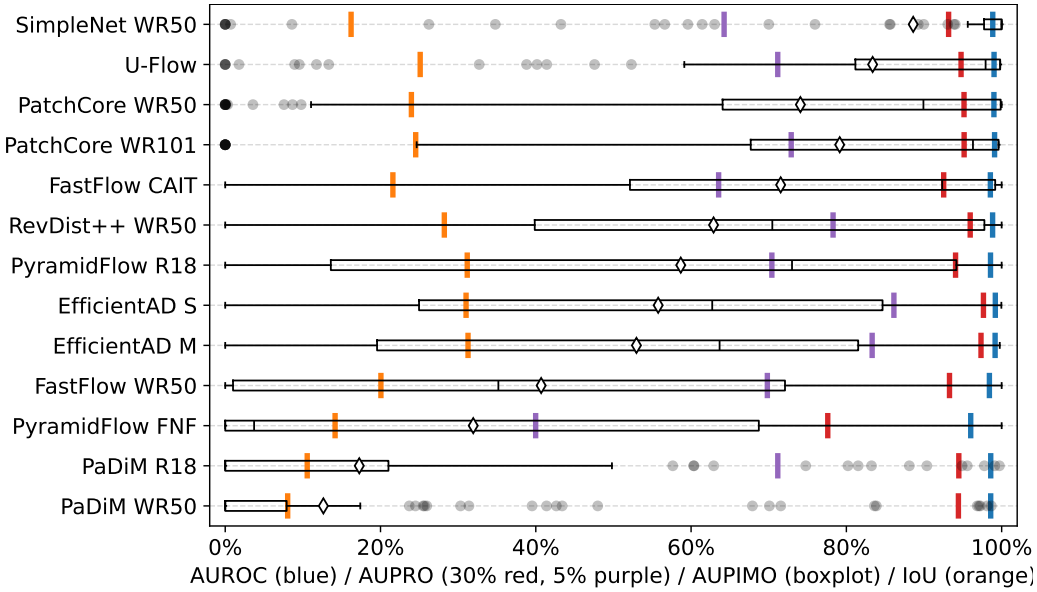


(c) Heatmaps. Images selected according to AUPIMO's statistics. Statistic and image index annotated on upper left corner. Image index annotated on upper left corner.

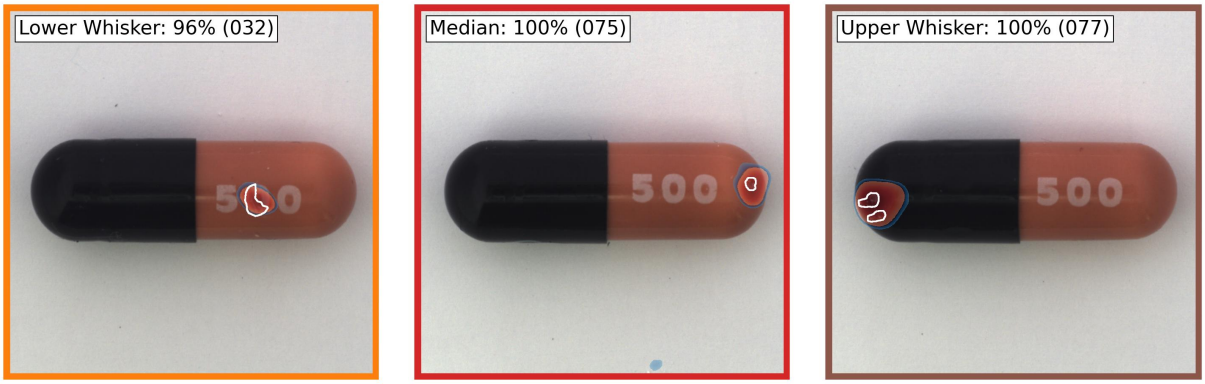
Figure A.5: Benchmark on MVTec AD / Cable. PIMO curves and heatmaps are from PatchCore WR101. 150 images (058 normal, 092 anomalous).



(a) Average rank diagram.

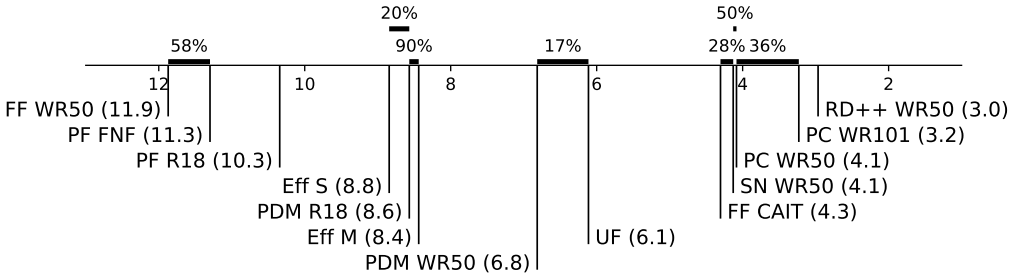


(b) Score distributions.

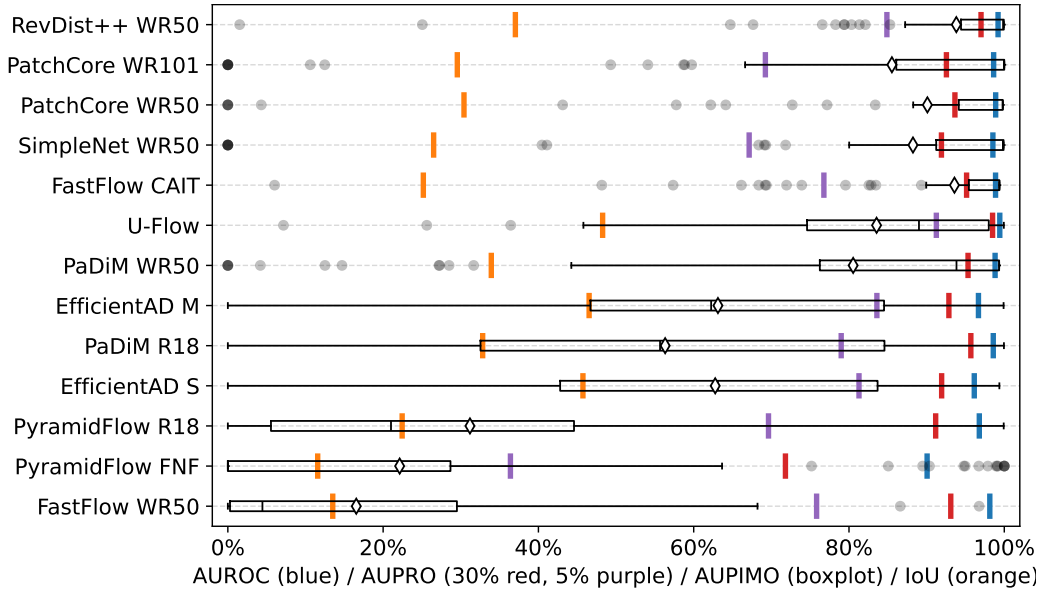


(c) Heatmaps. Images selected according to AUPIMO's statistics. Statistic and image index annotated on upper left corner.

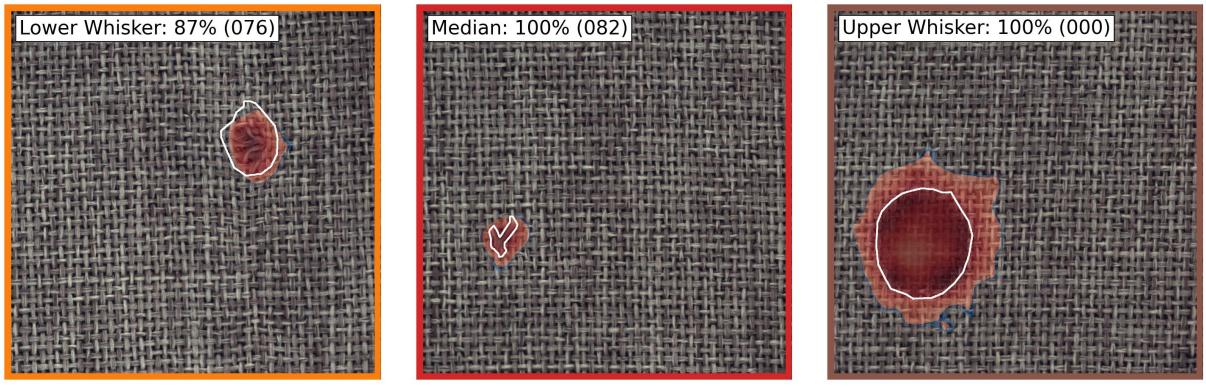
Figure A.6: Benchmark on MVTec AD / Capsule. PIMO curves and heatmaps are from SimpleNet WR50. 132 images (023 normal, 109 anomalous).



(a) Average rank diagram.

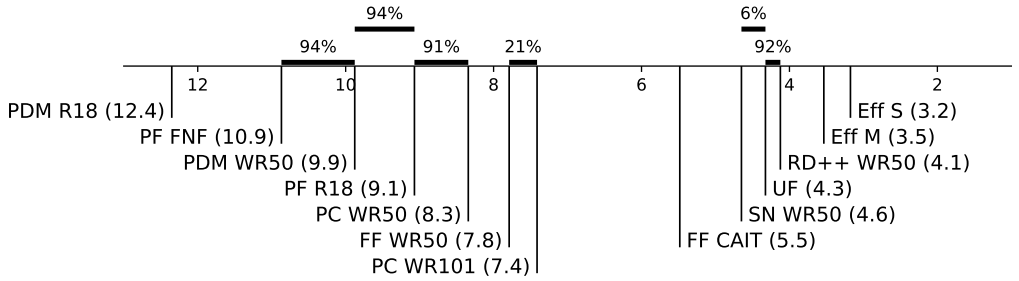


(b) Score distributions.

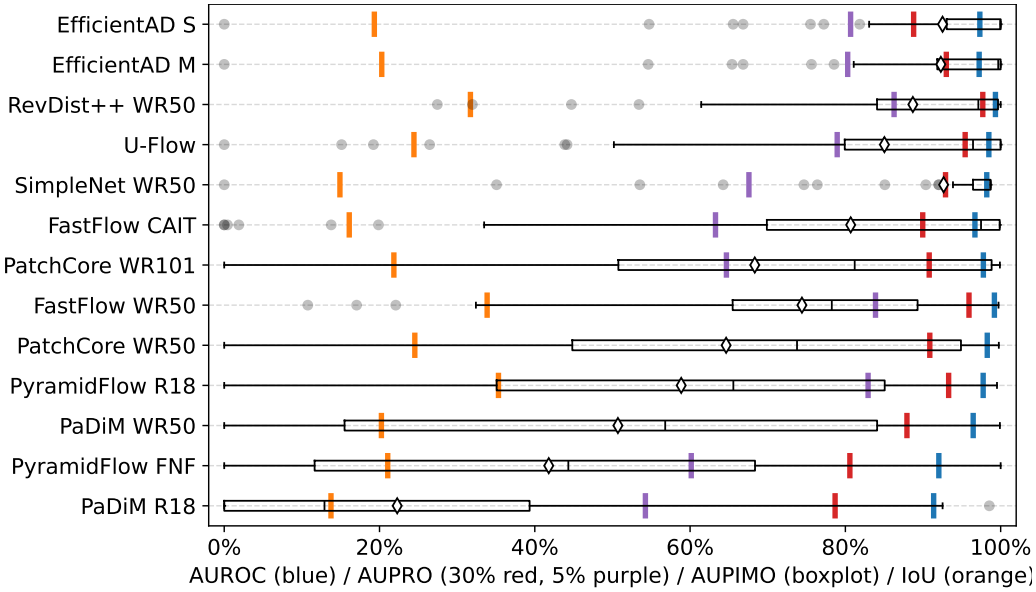


(c) Heatmaps. Images selected according to AUPIMO's statistics. Statistic and image index annotated on upper left corner.

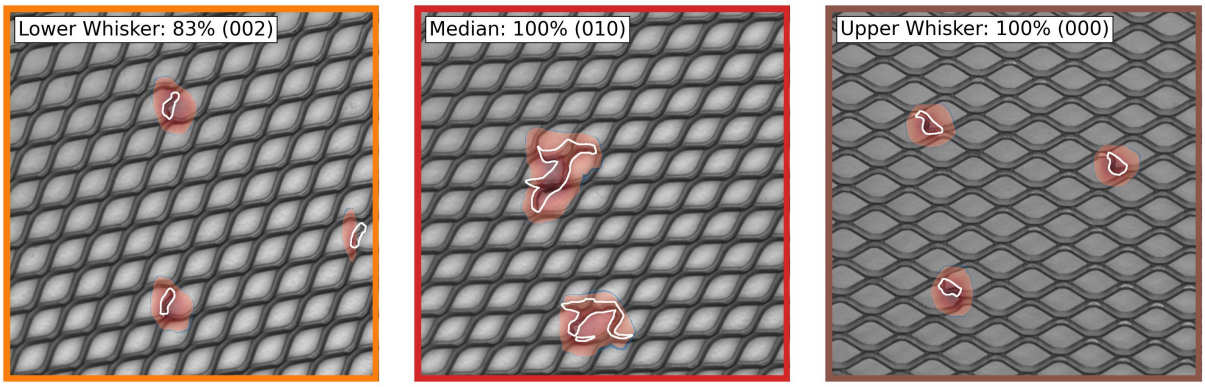
Figure A.7: Benchmark on MVTec AD / Carpet. PIMO curves and heatmaps are from RevDist++ WR50. 117 images (028 normal, 089 anomalous).



(a) Average rank diagram.

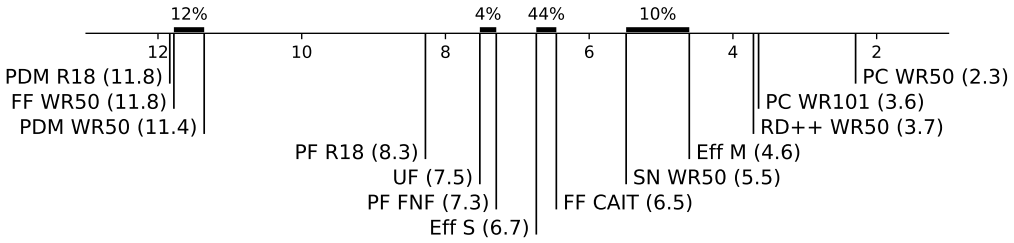


(b) Score distributions.

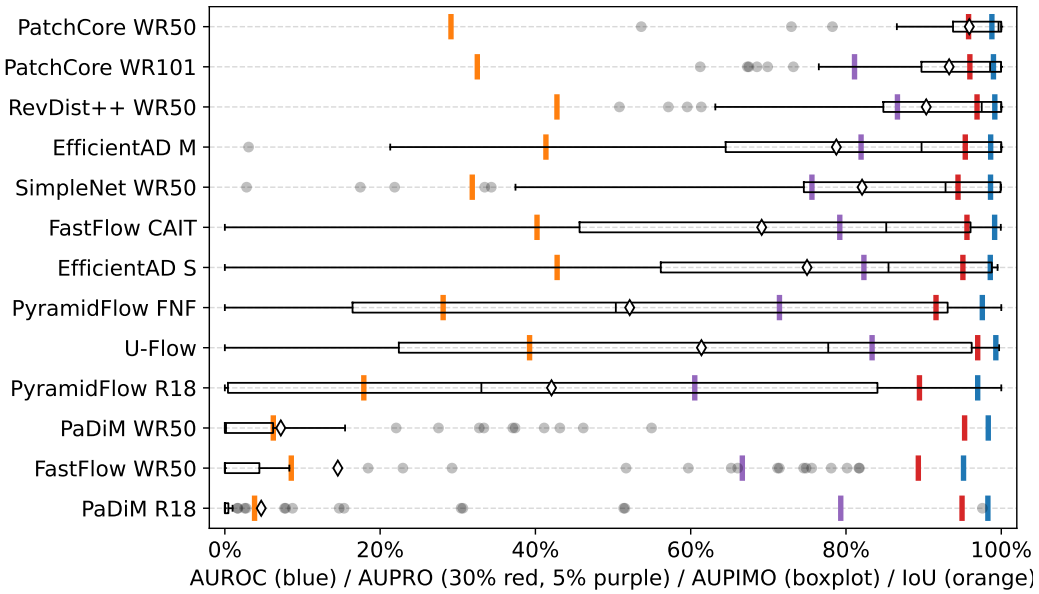


(c) Heatmaps. Images selected according to AUPIMO's statistics. Statistic and image index annotated on upper left corner.

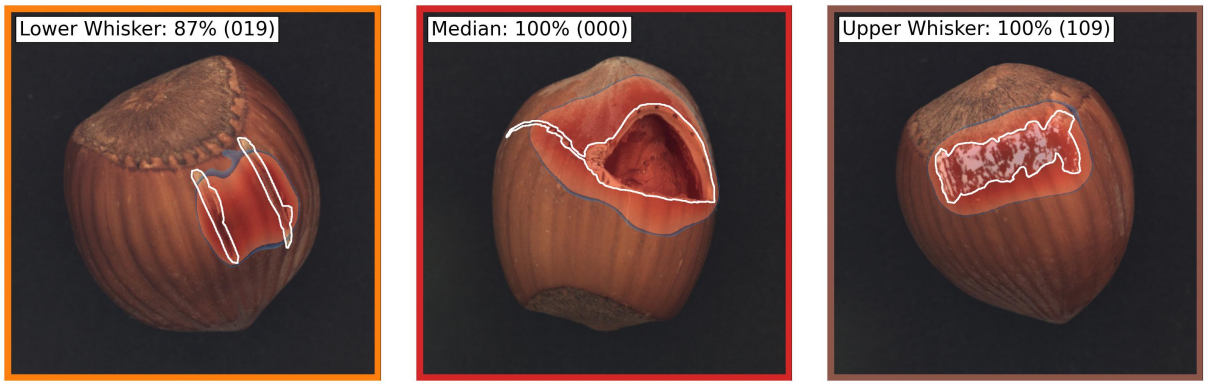
Figure A.8: Benchmark on MVTec AD / Grid. PIMO curves and heatmaps are from EfficientAD S. 078 images (021 normal, 057 anomalous).



(a) Average rank diagram.

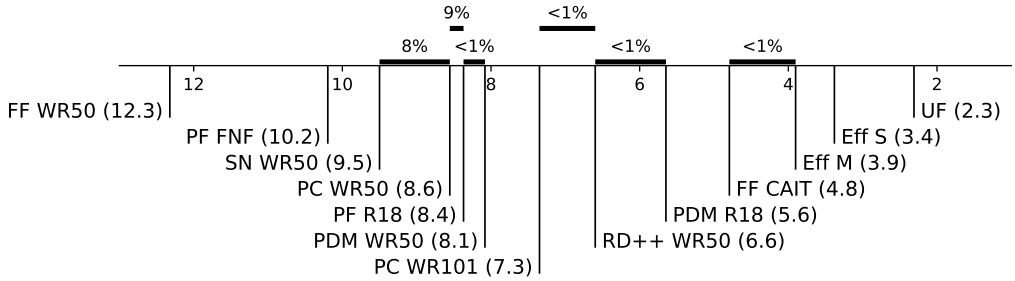


(b) Score distributions.

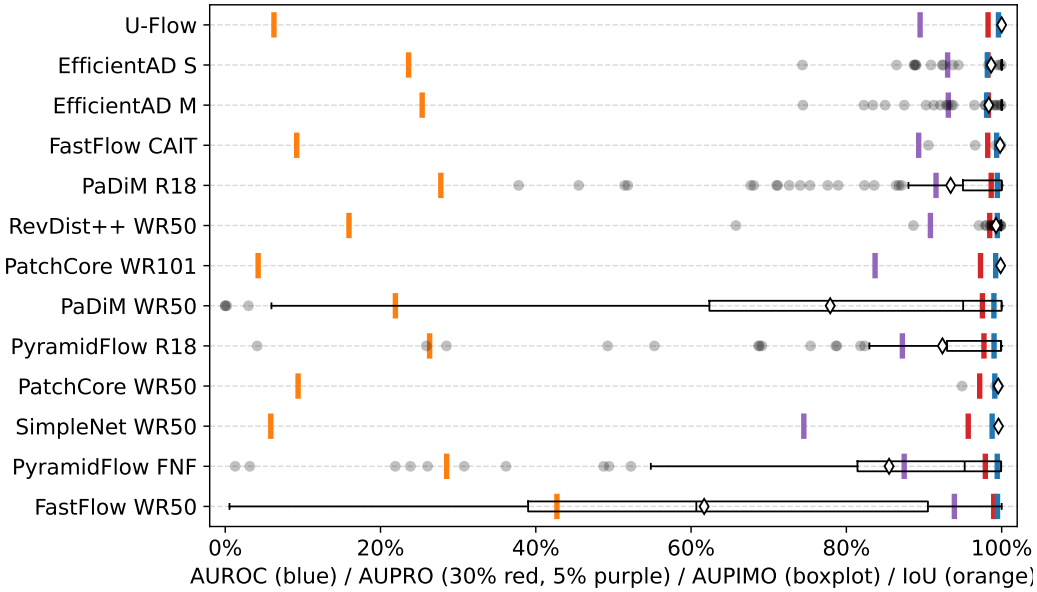


(c) Heatmaps. Images selected according to AUPIMO's statistics. Statistic and image index annotated on upper left corner.

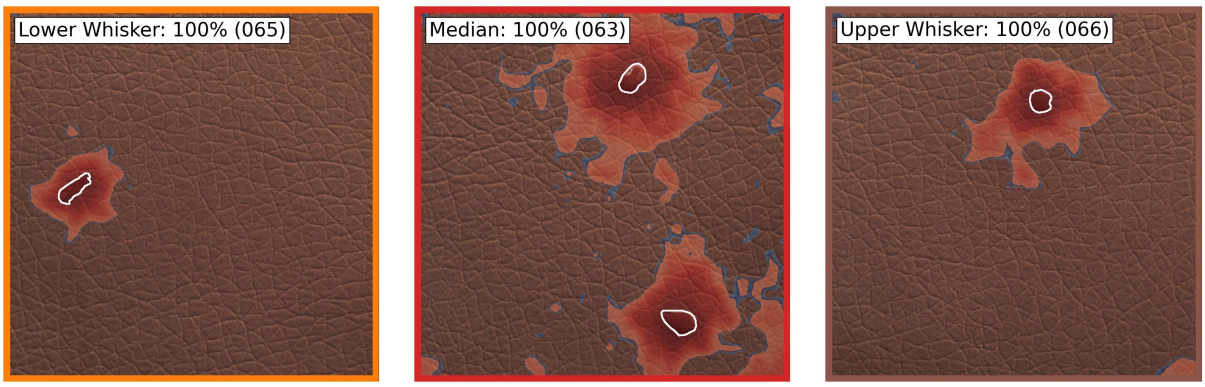
Figure A.9: Benchmark on MVTec AD / Hazelnut. PIMO curves and heatmaps are from PatchCore WR50. 110 images (040 normal, 070 anomalous).



(a) Average rank diagram.

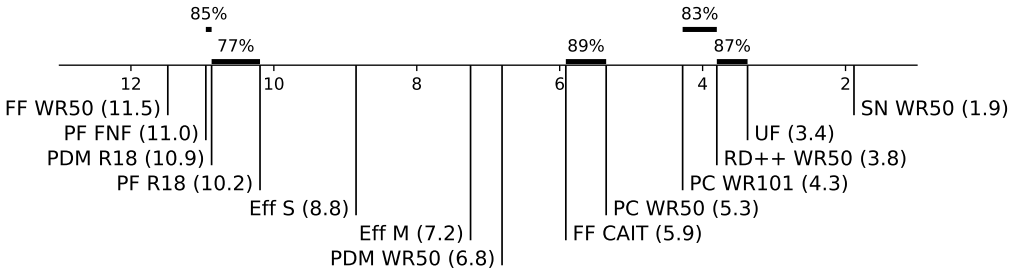


(b) Score distributions.

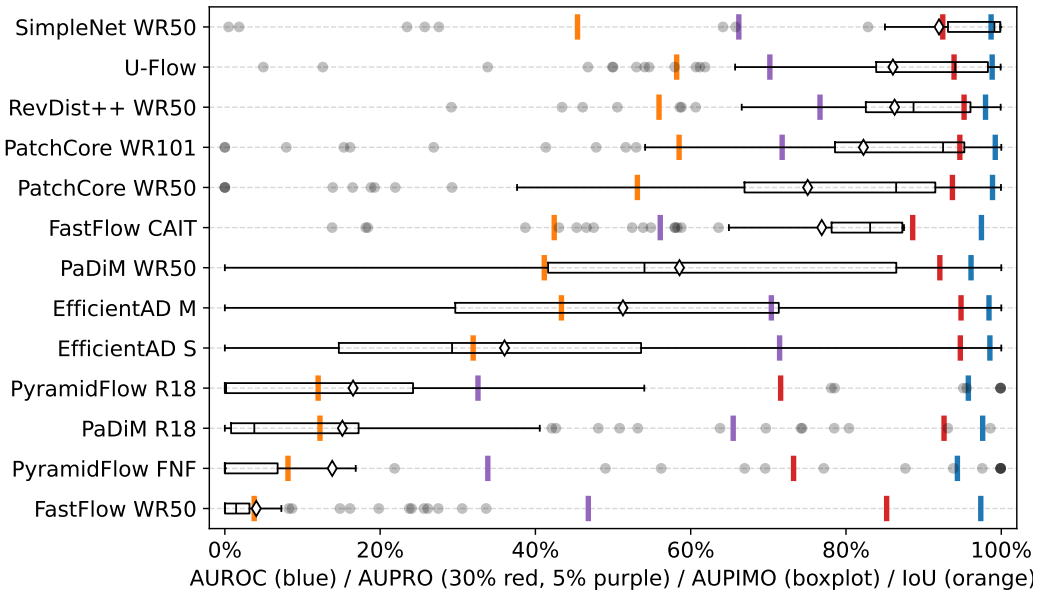


(c) Heatmaps. Images selected according to AUPIMO's statistics. Statistic and image index annotated on upper left corner.

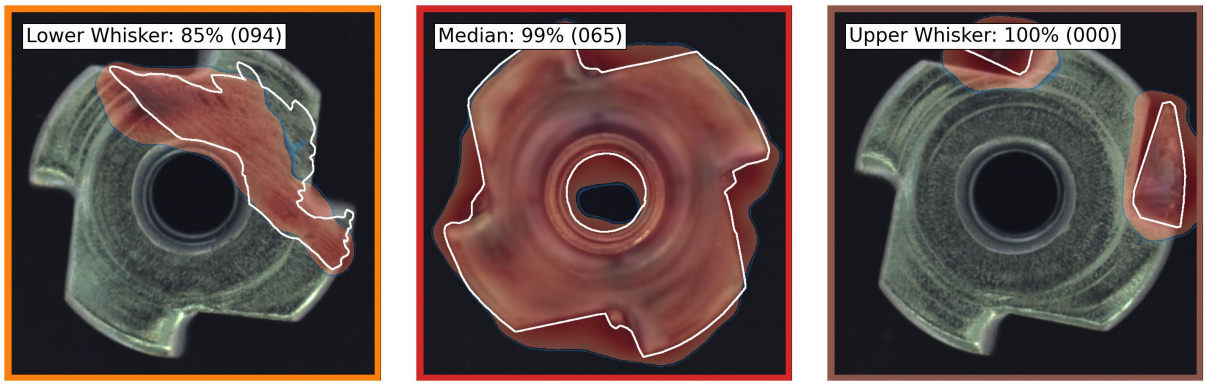
Figure A.10: Benchmark on MVTec AD / Leather. PIMO curves and heatmaps are from U-Flow. 124 images (032 normal, 092 anomalous).



(a) Average rank diagram.

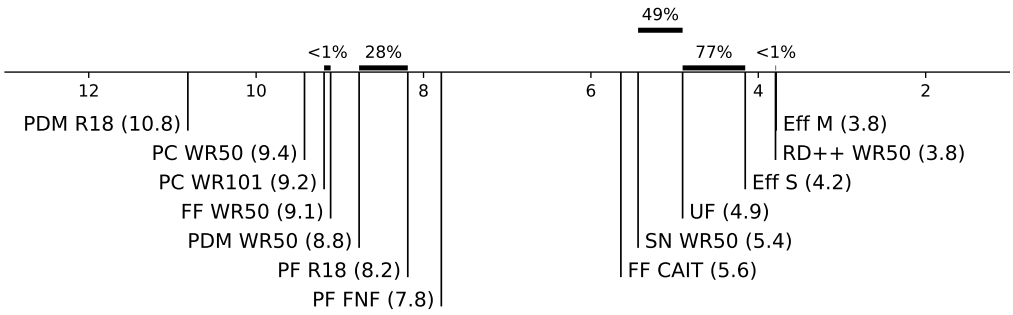


(b) Score distributions.

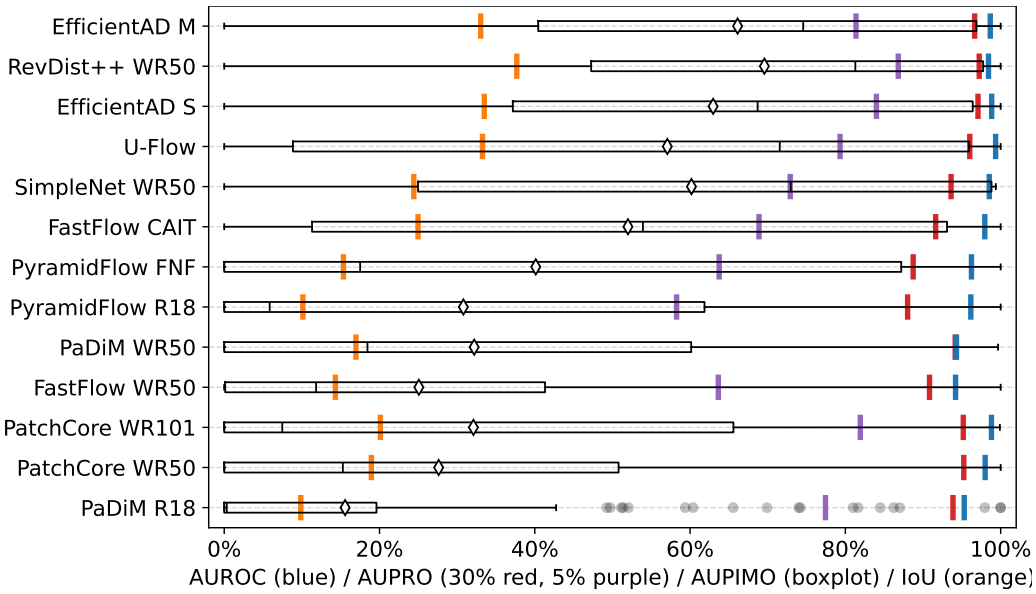


(c) Heatmaps. Images selected according to AUPIMO's statistics. Statistic and image index annotated on upper left corner.

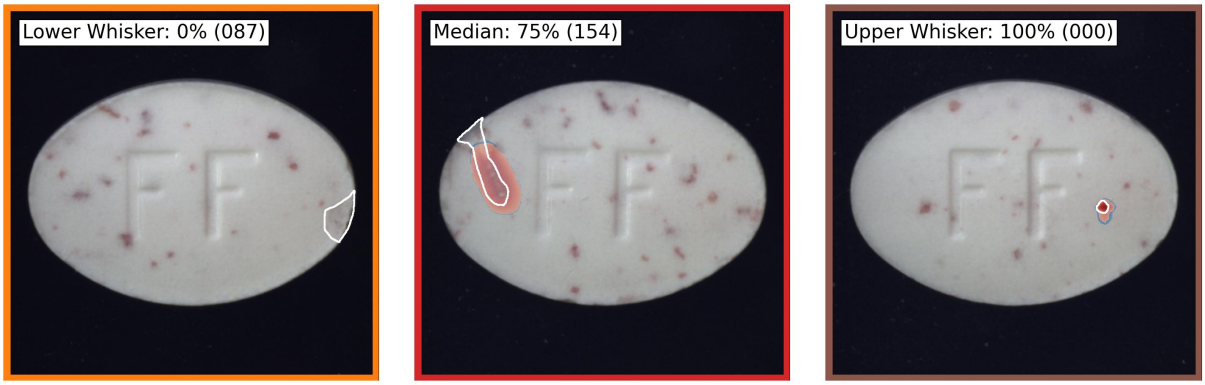
Figure A.11: Benchmark on MVTec AD / Metal Nut. PIMO curves and heatmaps are from SimpleNet WR50. 115 images (022 normal, 093 anomalous).



(a) Average rank diagram.

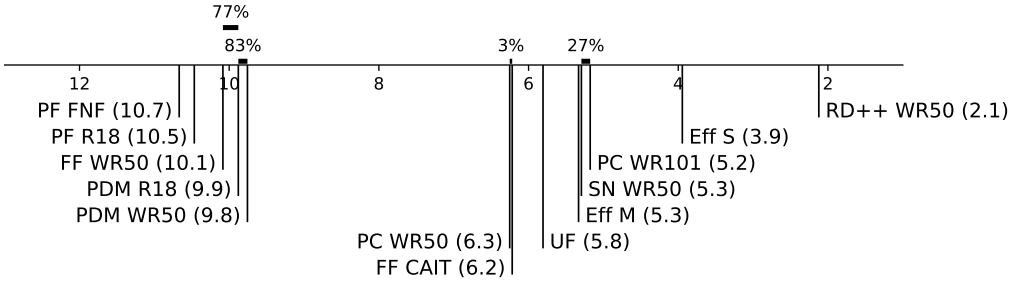


(b) Score distributions.

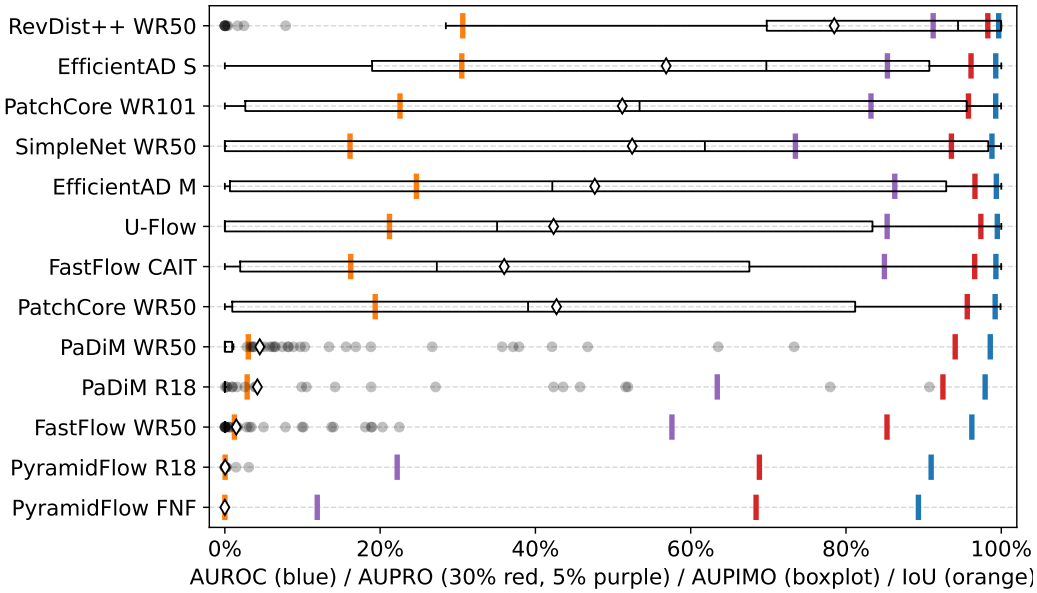


(c) Heatmaps. Images selected according to AUPIMO's statistics. Statistic and image index annotated on upper left corner.

Figure A.12: Benchmark on MVTec AD / Pill. PIMO curves and heatmaps are from EfficientAD M. 167 images (026 normal, 141 anomalous).



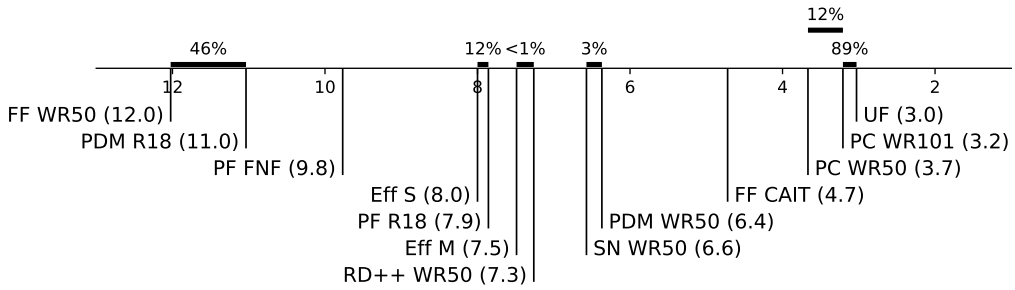
(a) Average rank diagram.



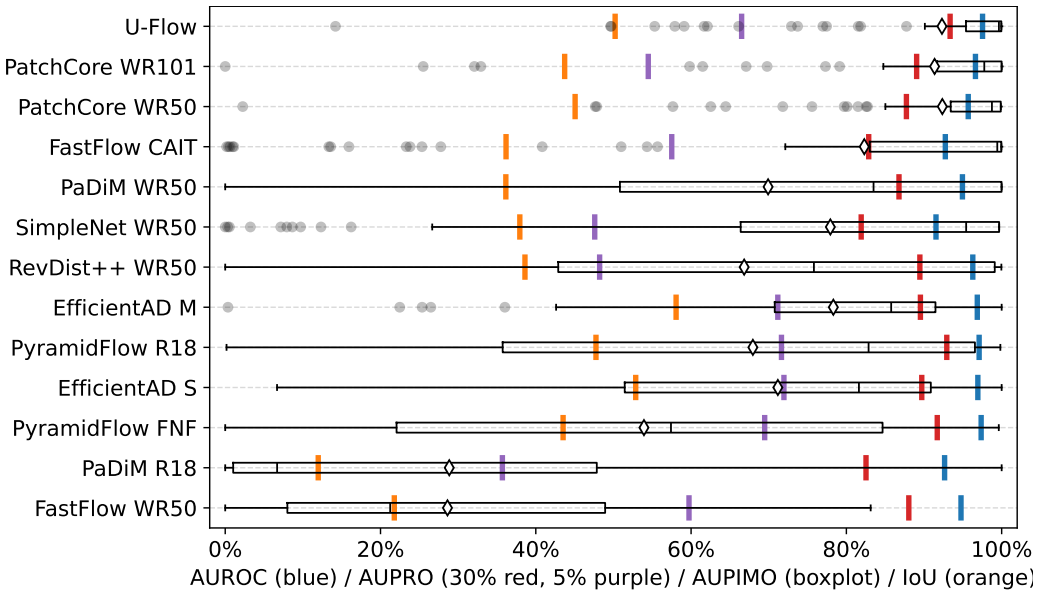
(b) Score distributions.

(c) Heatmaps. Images selected according to AUPIMO's statistics. Statistic and image index annotated on upper left corner.

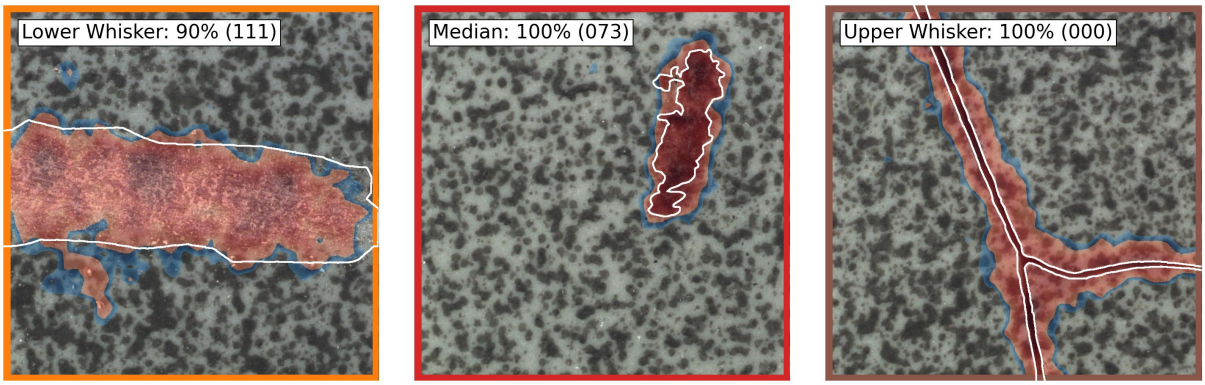
Figure A.13: Benchmark on MVTec AD / Screw. PIMO curves and heatmaps are from RevDist++ WR50. 160 images (041 normal, 119 anomalous).



(a) Average rank diagram.

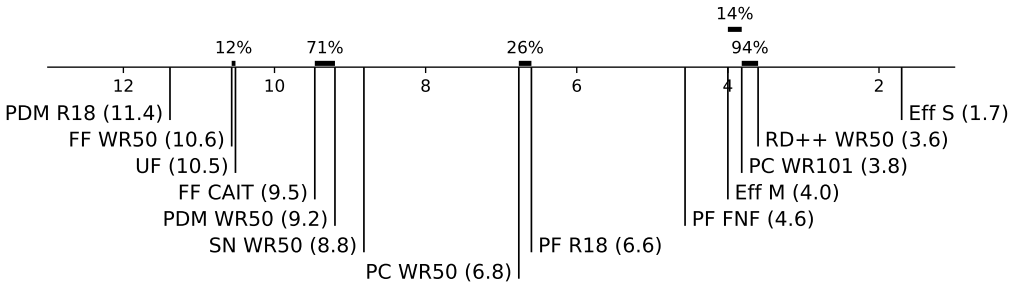


(b) Score distributions.

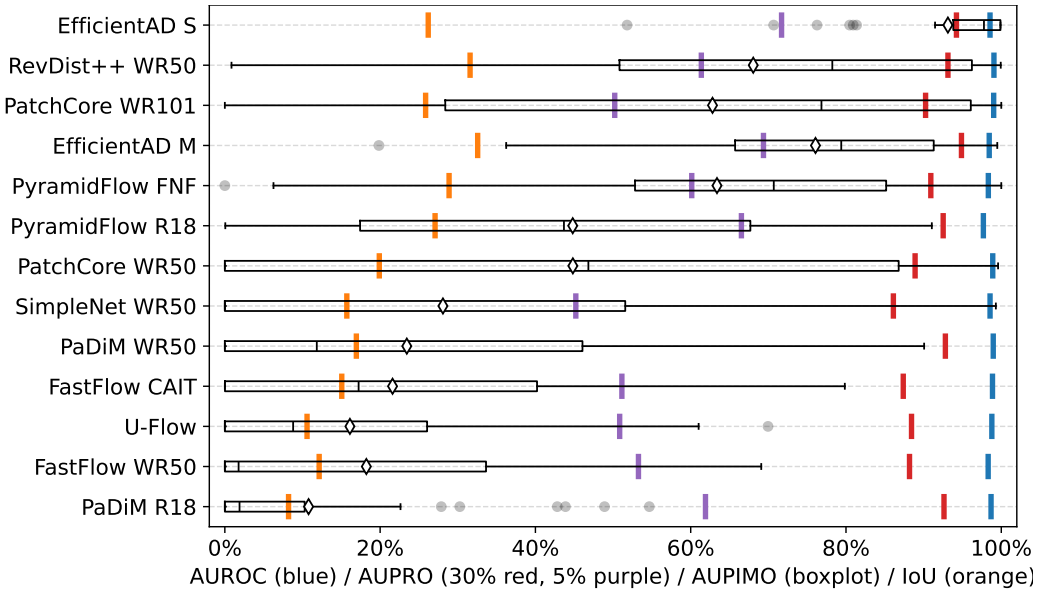


(c) Heatmaps. Images selected according to AUPIMO's statistics. Statistic and image index annotated on upper left corner.

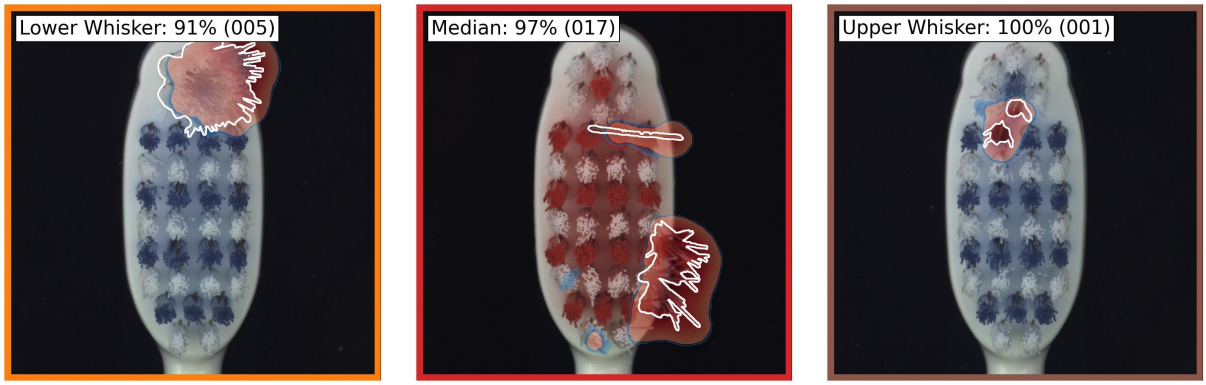
Figure A.14: Benchmark on MVTec AD / Tile. PIMO curves and heatmaps are from U-Flow. 117 images (033 normal, 084 anomalous).



(a) Average rank diagram.

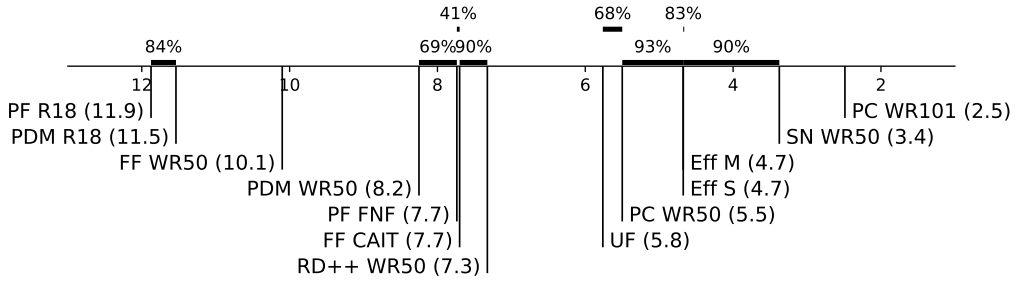


(b) Score distributions.

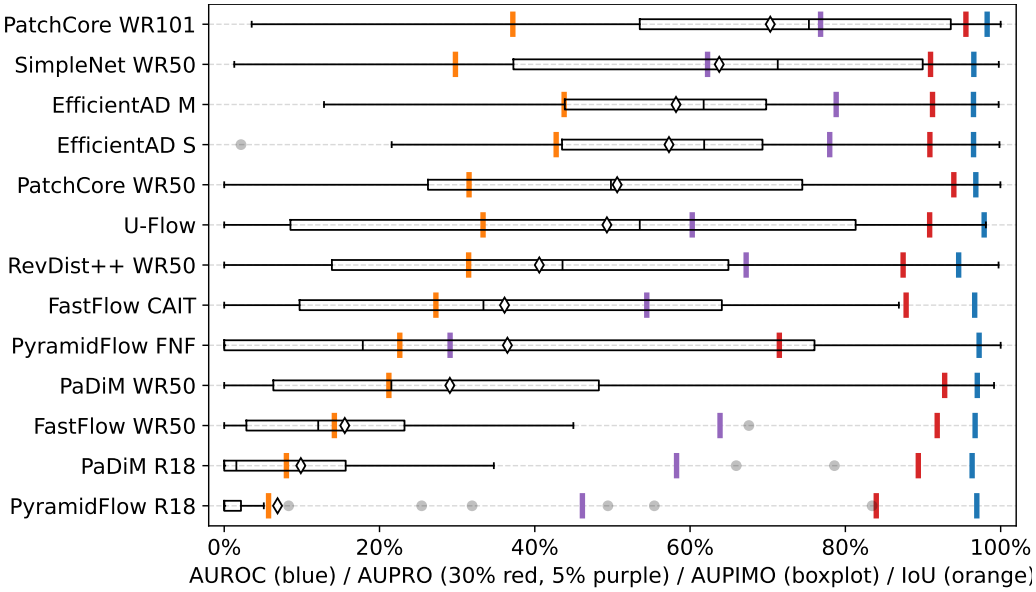


(c) Heatmaps. Images selected according to AUPIMO's statistics. Statistic and image index annotated on upper left corner.

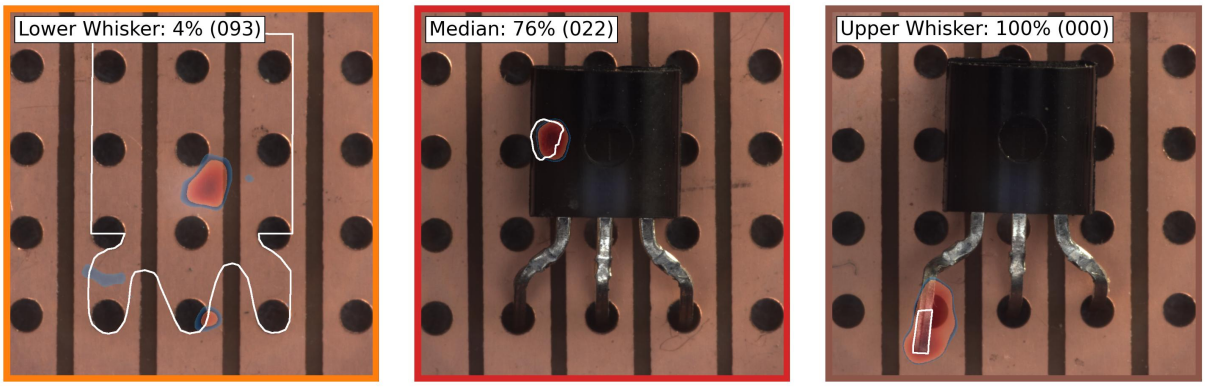
Figure A.15: Benchmark on MVTec AD / Toothbrush. PIMO curves and heatmaps are from EfficientAD S. 042 images (012 normal, 030 anomalous).



(a) Average rank diagram.

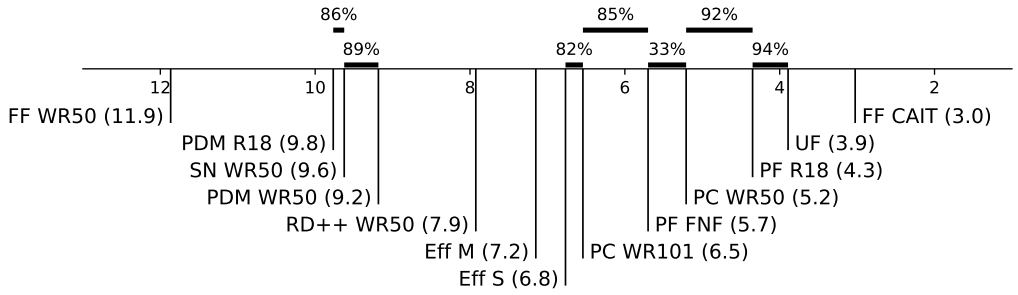


(b) Score distributions.

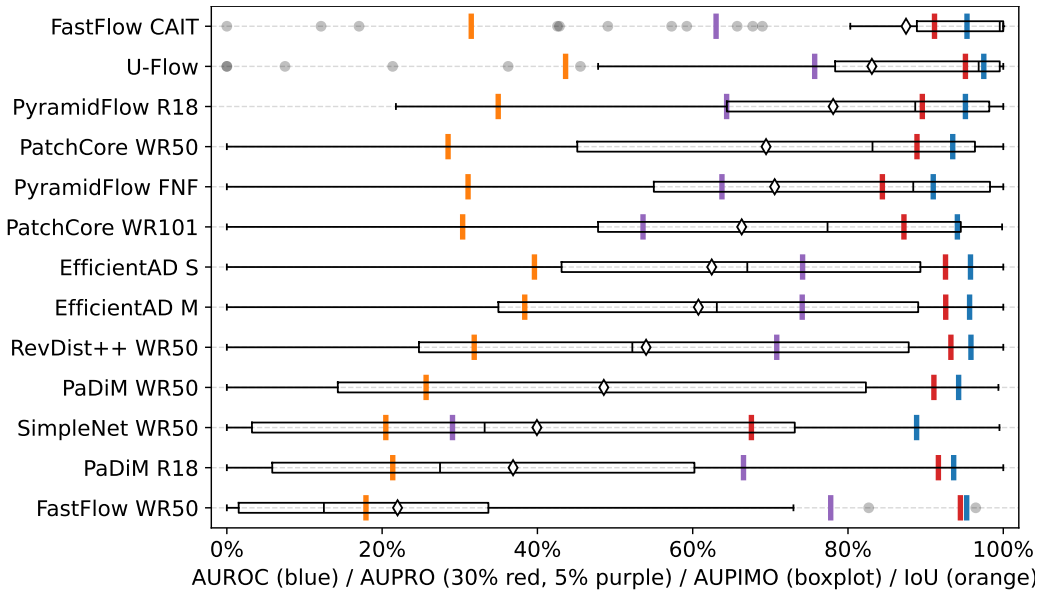


(c) Heatmaps. Images selected according to AUPIMO's statistics. Statistic and image index annotated on upper left corner.

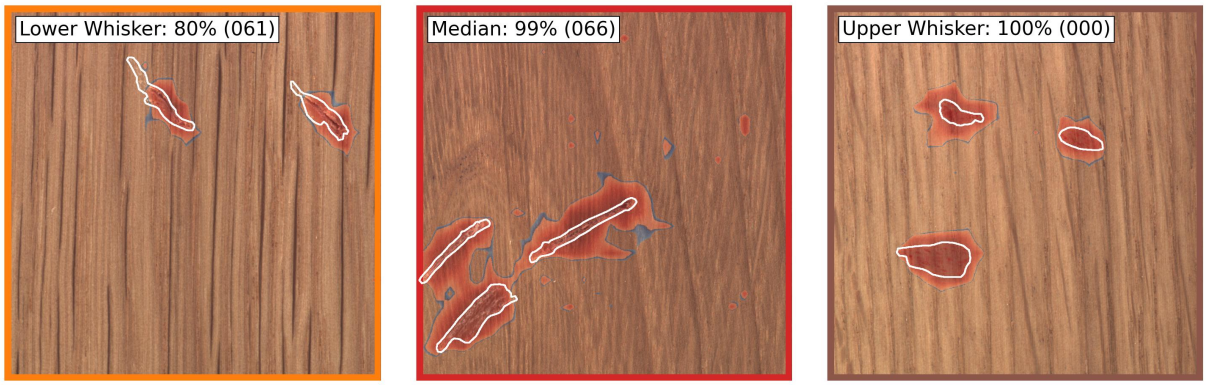
Figure A.16: Benchmark on MVTec AD / Transistor. PIMO curves and heatmaps are from PatchCore WR101. 100 images (060 normal, 040 anomalous).



(a) Average rank diagram.

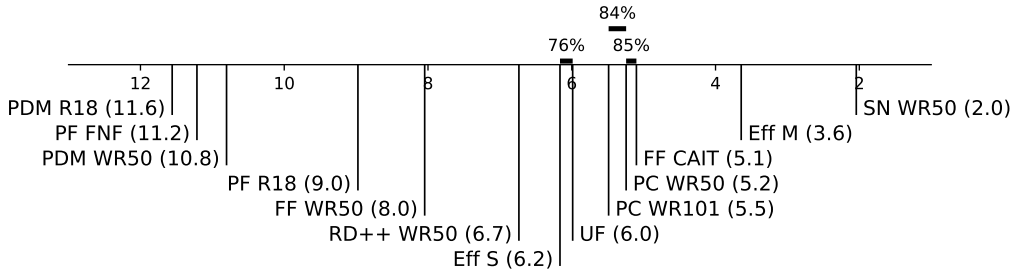


(b) Score distributions.

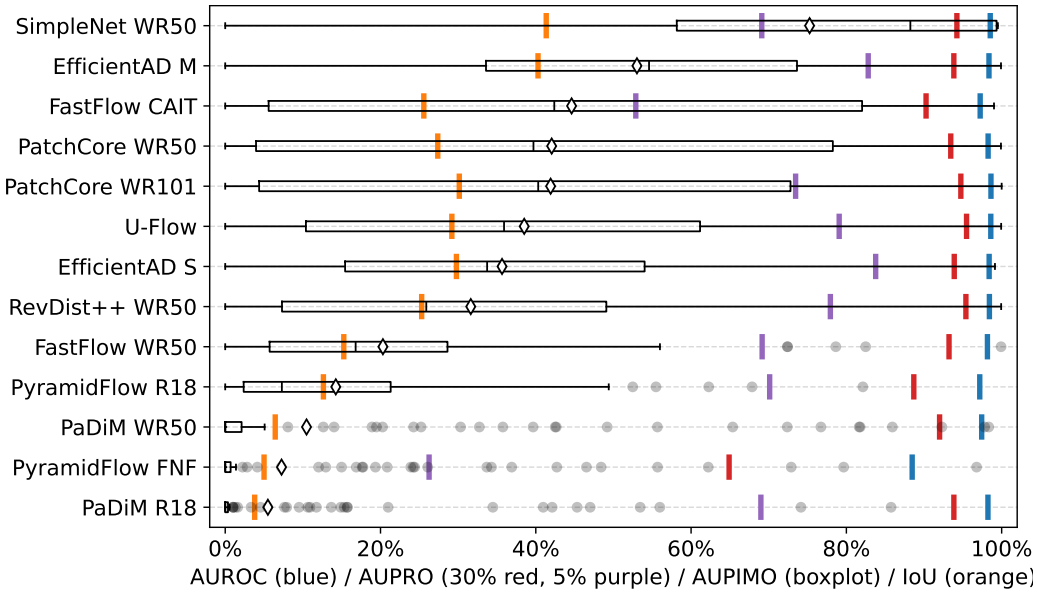


(c) Heatmaps. Images selected according to AUPIMO's statistics. Statistic and image index annotated on upper left corner.

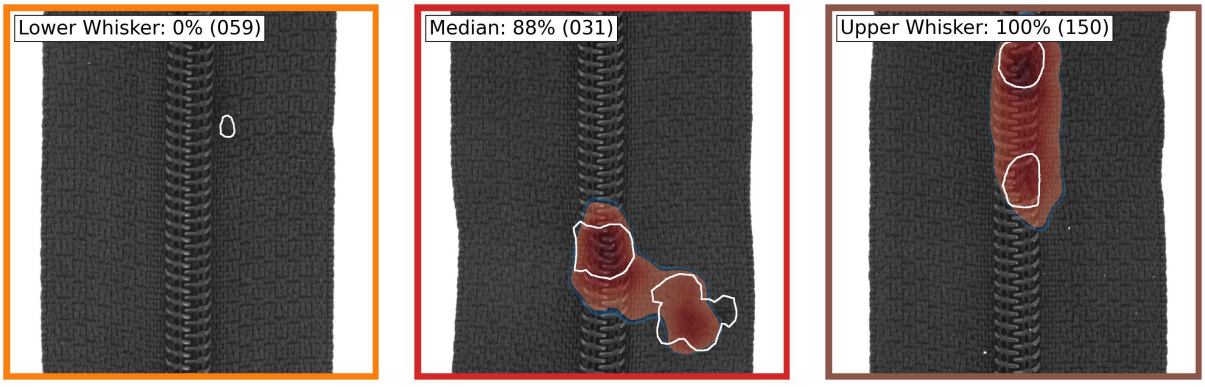
Figure A.17: Benchmark on MVTec AD / Wood. PIMO curves and heatmaps are from FastFlow CAIT. 079 images (019 normal, 060 anomalous).



(a) Average rank diagram.

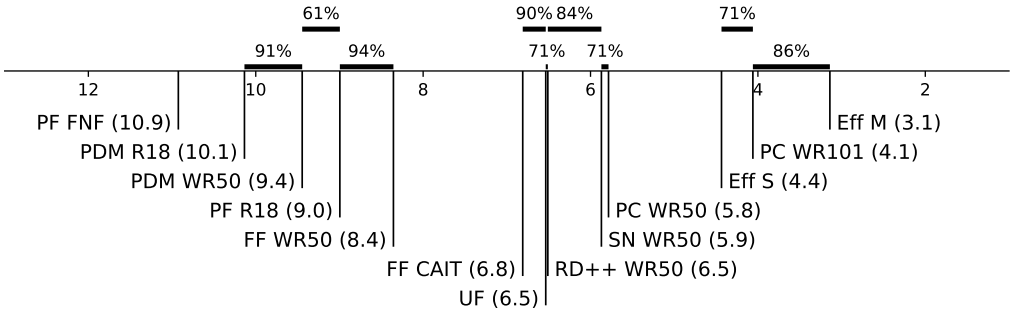


(b) Score distributions.

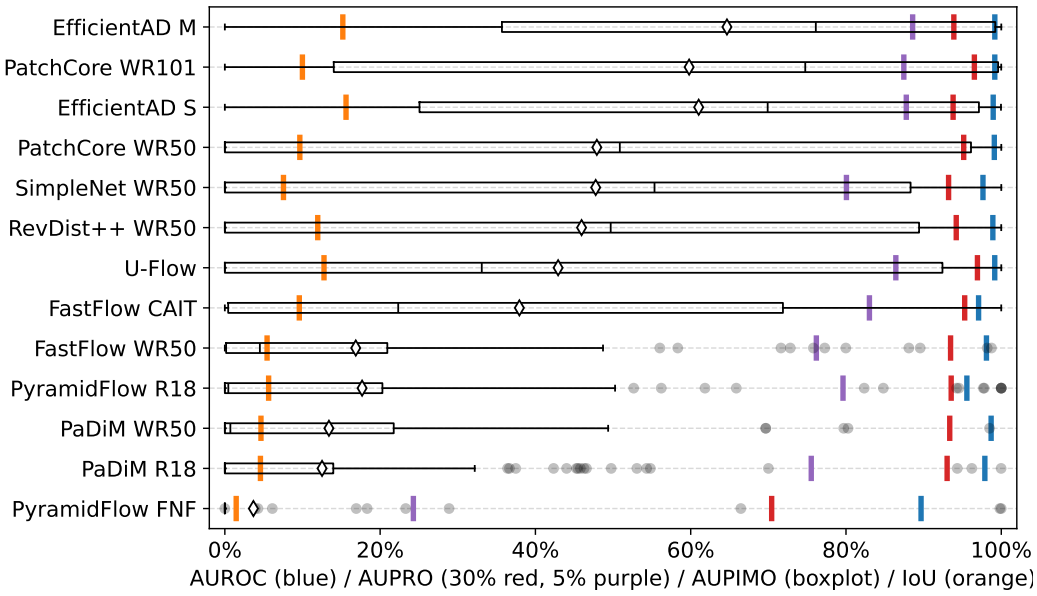


(c) Heatmaps. Images selected according to AUPIMO's statistics. Statistic and image index annotated on upper left corner.

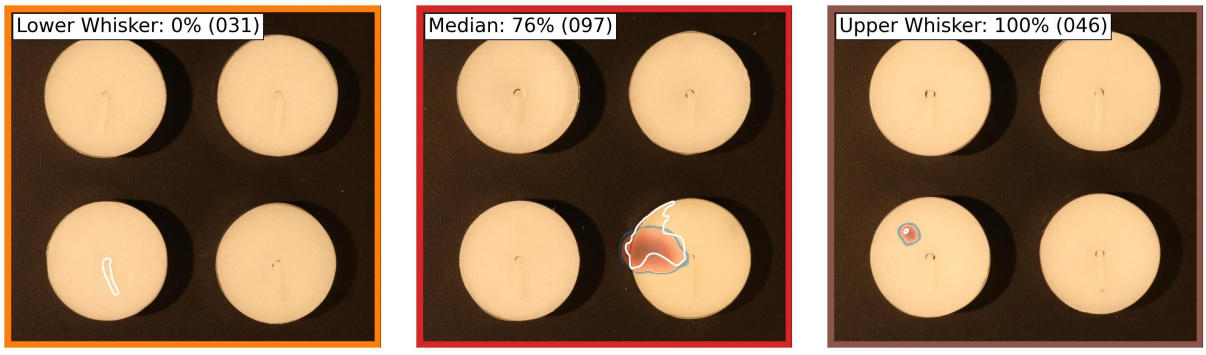
Figure A.18: Benchmark on MVTec AD / Zipper. PIMO curves and heatmaps are from SimpleNet WR50. 151 images (032 normal, 119 anomalous).



(a) Average rank diagram.

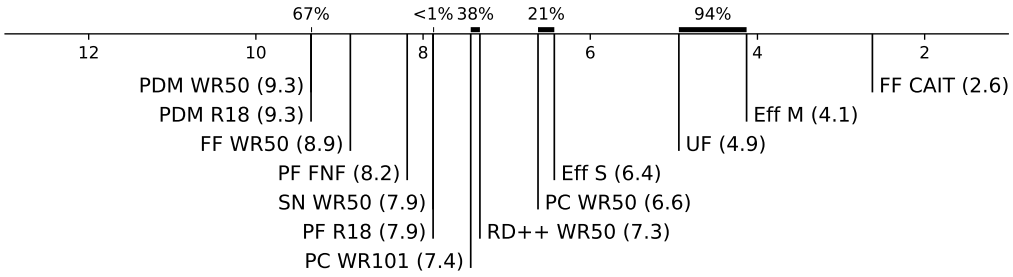


(b) Score distributions.

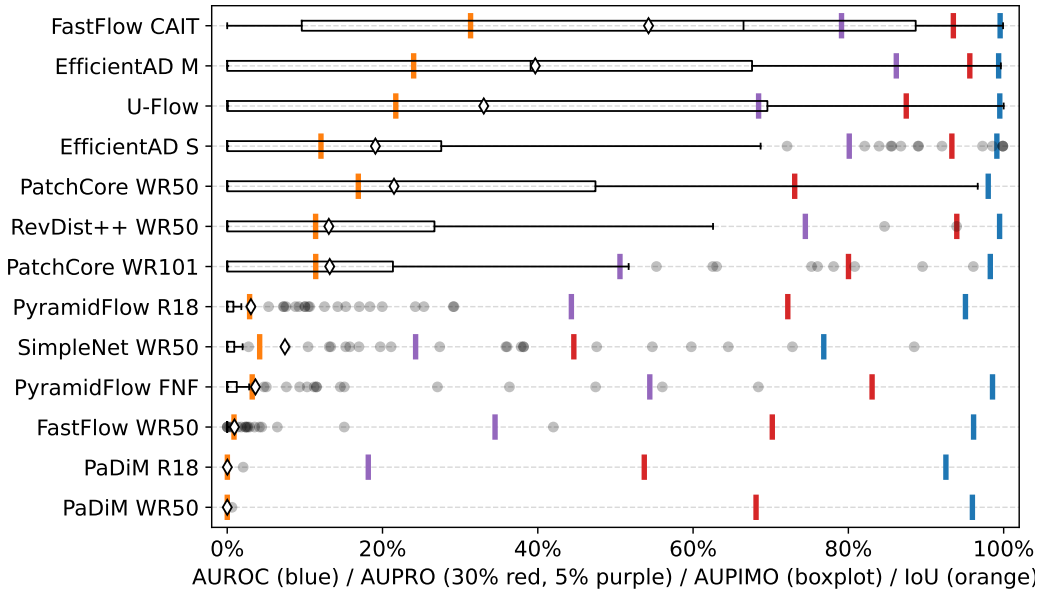


(c) Heatmaps. Images selected according to AUPIMO's statistics. Statistic and image index annotated on upper left corner.

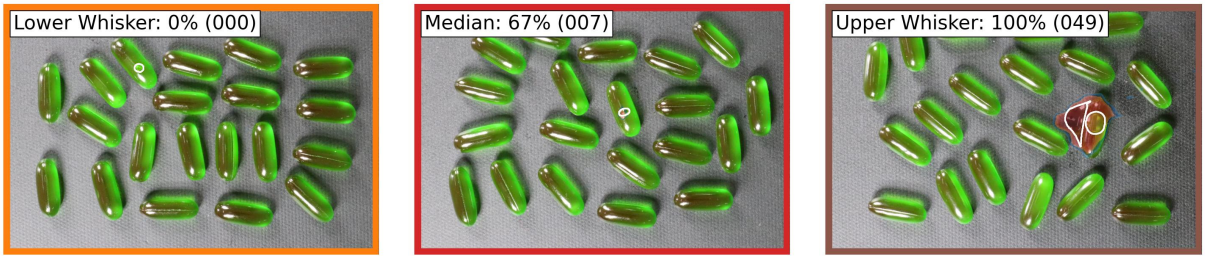
Figure A.19: Benchmark on VisA / Candle. PIMO curves and heatmaps are from EfficientAD M. 200 images (100 normal, 100 anomalous).



(a) Average rank diagram.

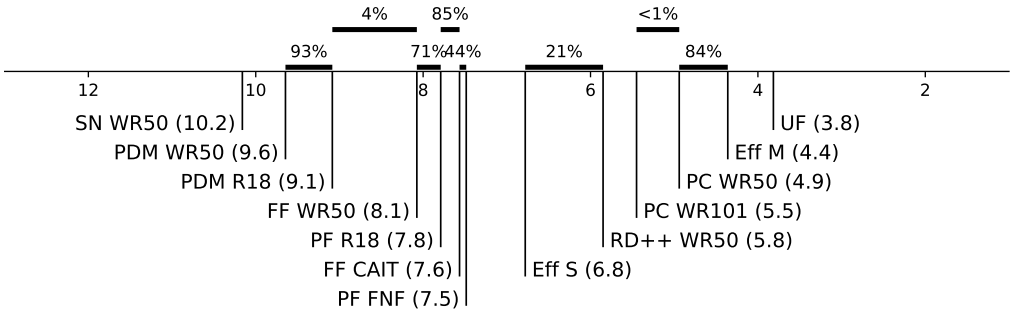


(b) Score distributions.

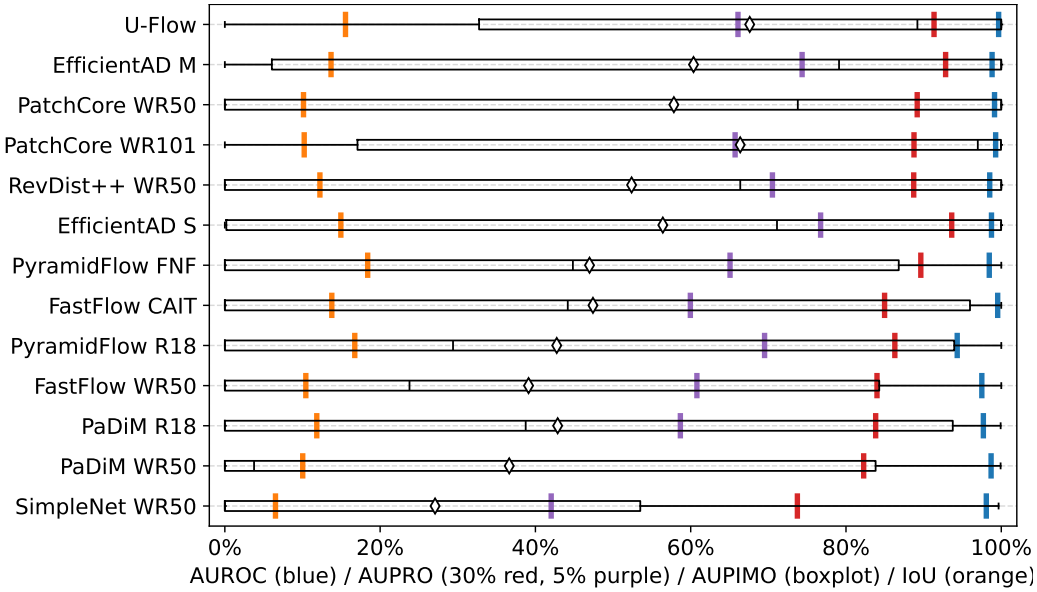


(c) Heatmaps. Images selected according to AUPIMO's statistics. Statistic and image index annotated on upper left corner.

Figure A.20: Benchmark on VisA / Capsules. PIMO curves and heatmaps are from FastFlow CAIT. 160 images (060 normal, 100 anomalous).



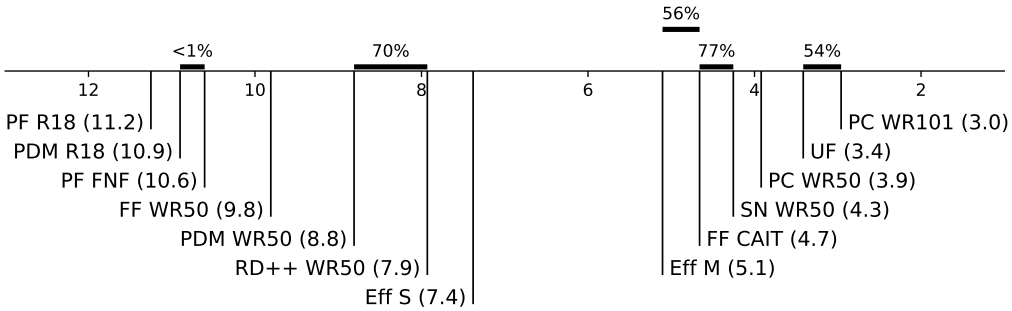
(a) Average rank diagram.



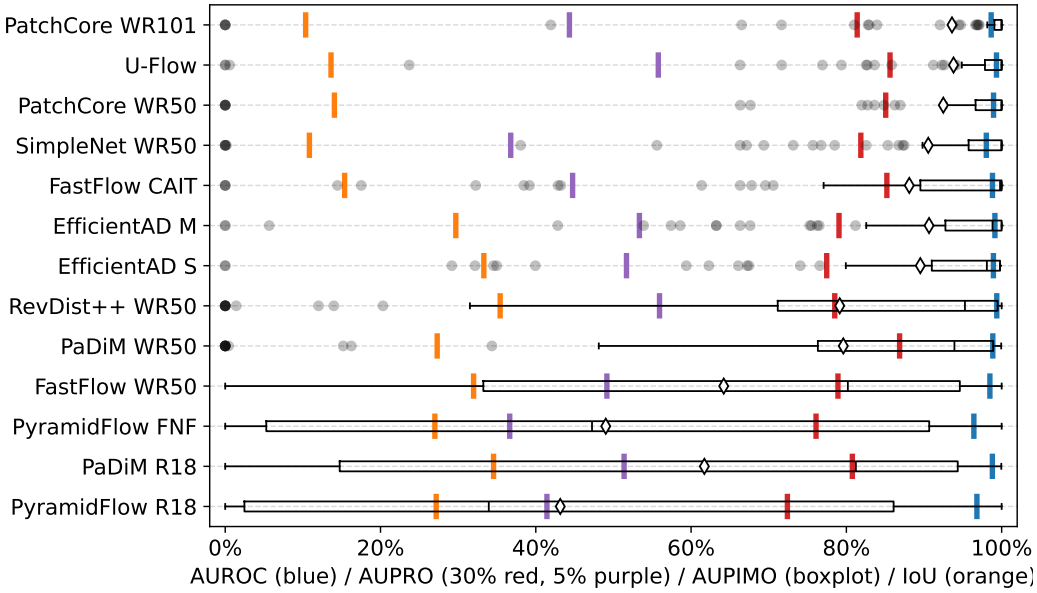
(b) Score distributions.

(c) Heatmaps. Images selected according to AUPIMO's statistics. Statistic and image index annotated on upper left corner.

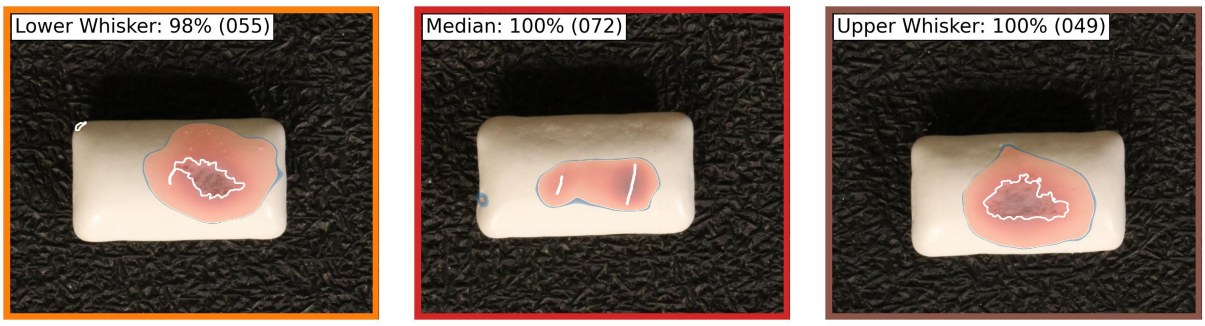
Figure A.21: Benchmark on VisA / Cashew. PIMO curves and heatmaps are from U-Flow. 150 images (050 normal, 100 anomalous).



(a) Average rank diagram.

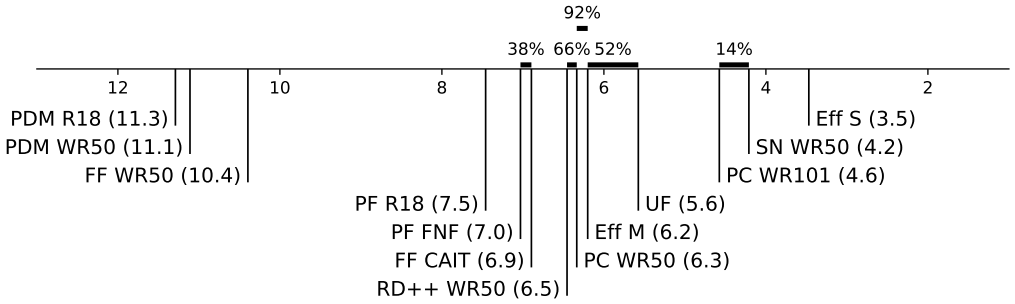


(b) Score distributions.

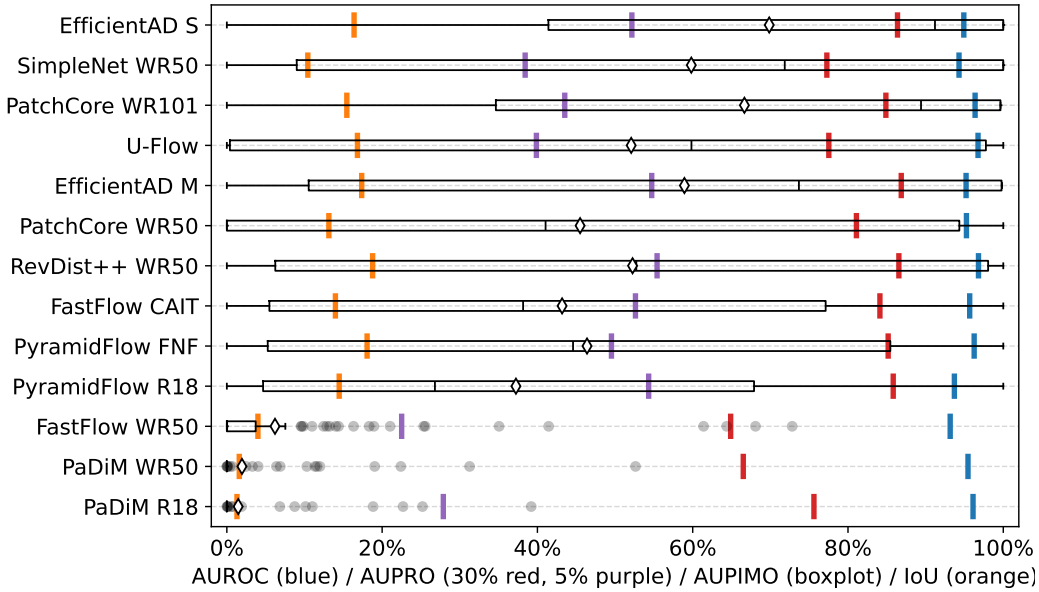


(c) Heatmaps. Images selected according to AUPIMO's statistics. Statistic and image index annotated on upper left corner.

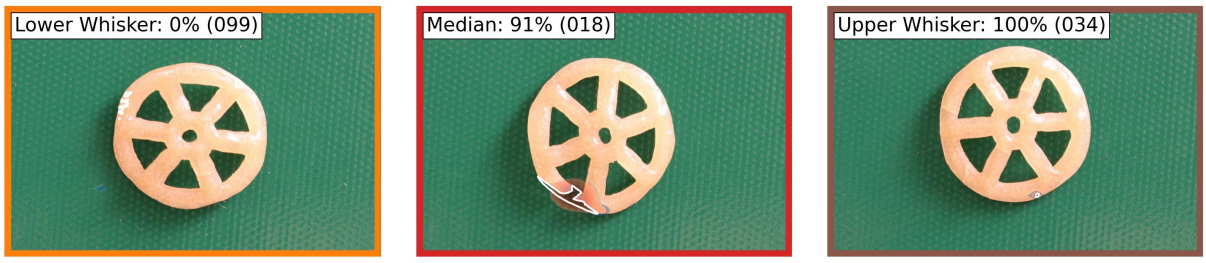
Figure A.22: Benchmark on VisA / Chewing Gum. PIMO curves and heatmaps are from PatchCore WR101. 150 images (050 normal, 100 anomalous).



(a) Average rank diagram.

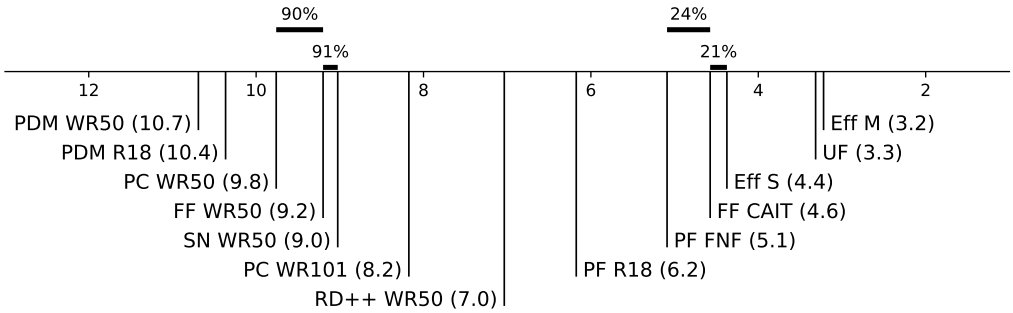


(b) Score distributions.

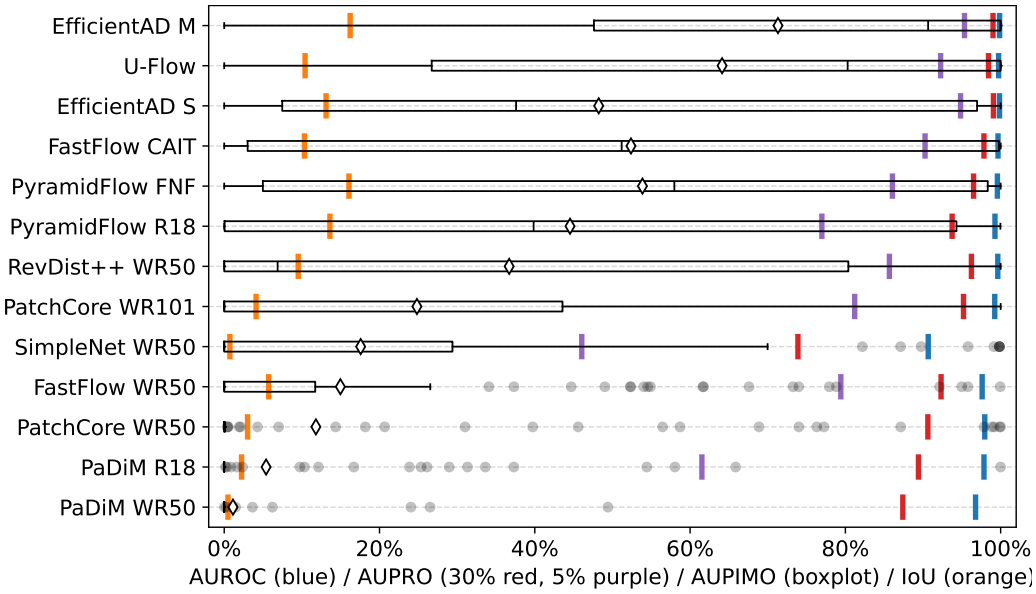


(c) Heatmaps. Images selected according to AUPIMO's statistics. Statistic and image index annotated on upper left corner.

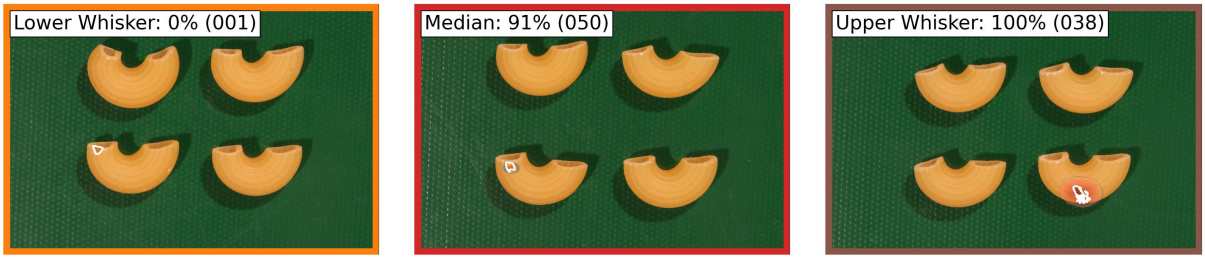
Figure A.23: Benchmark on VisA / Fryum. PIMO curves and heatmaps are from EfficientAD S. 150 images (050 normal, 100 anomalous).



(a) Average rank diagram.

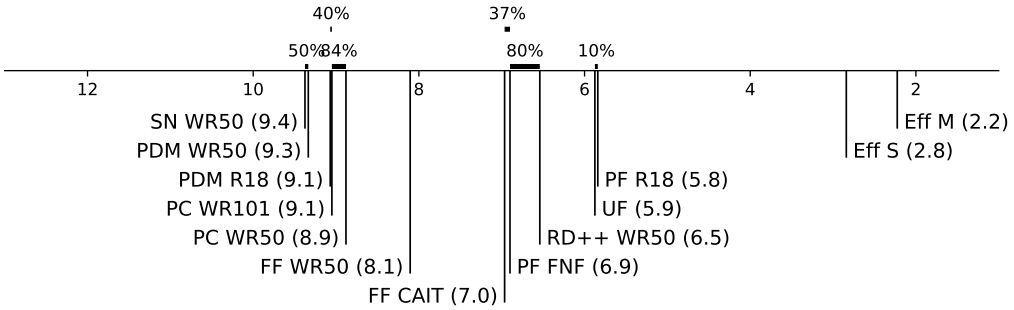


(b) Score distributions.

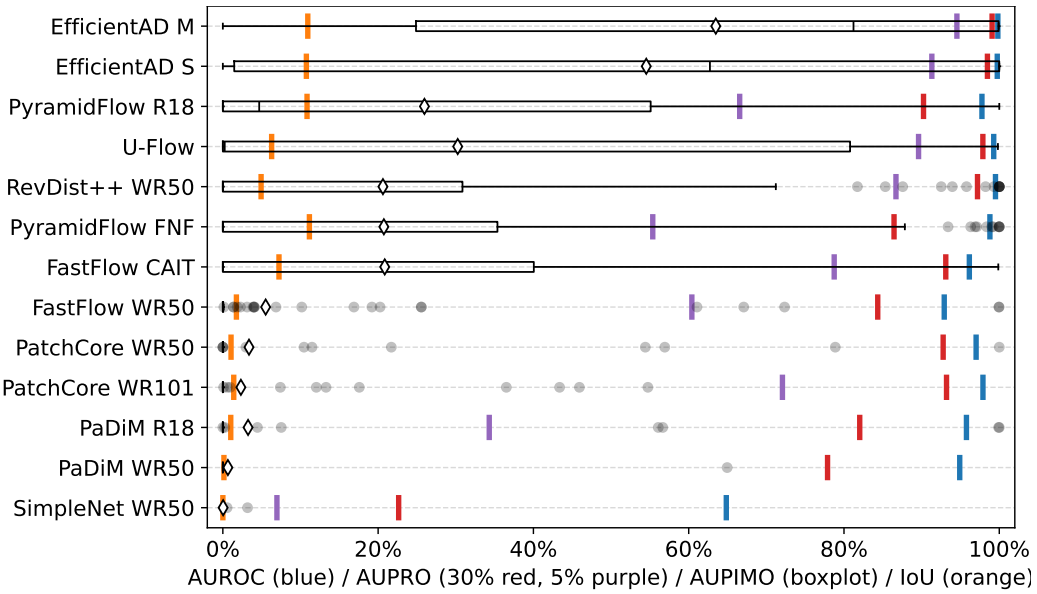


(c) Heatmaps. Images selected according to AUPIMO's statistics. Statistic and image index annotated on upper left corner.

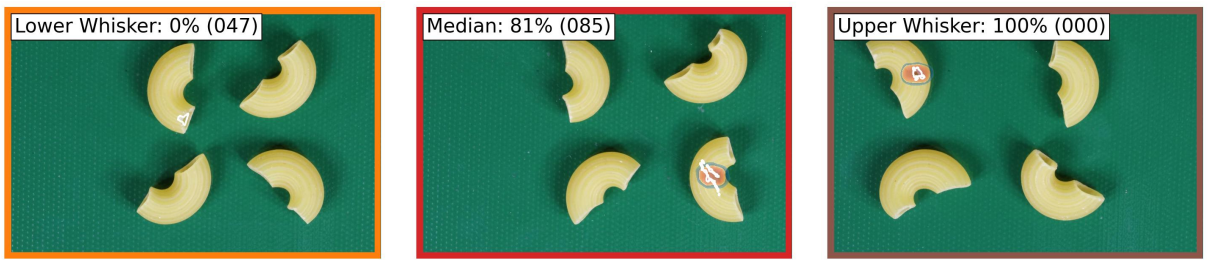
Figure A.24: Benchmark on VisA / Macaroni 1. PIMO curves and heatmaps are from EfficientAD M. 200 images (100 normal, 100 anomalous).



(a) Average rank diagram.

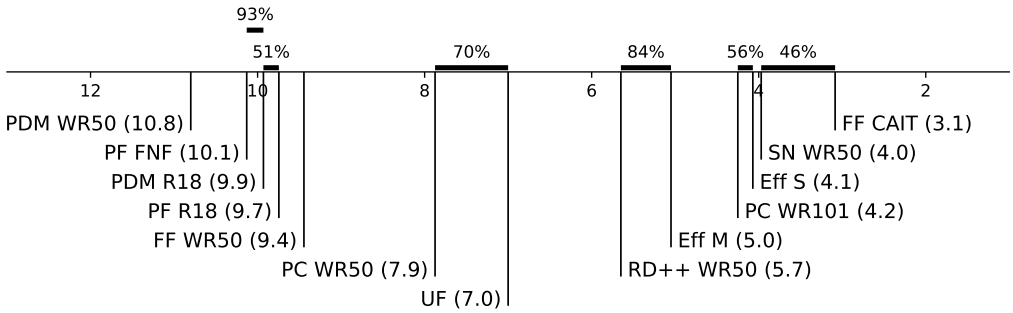


(b) Score distributions.

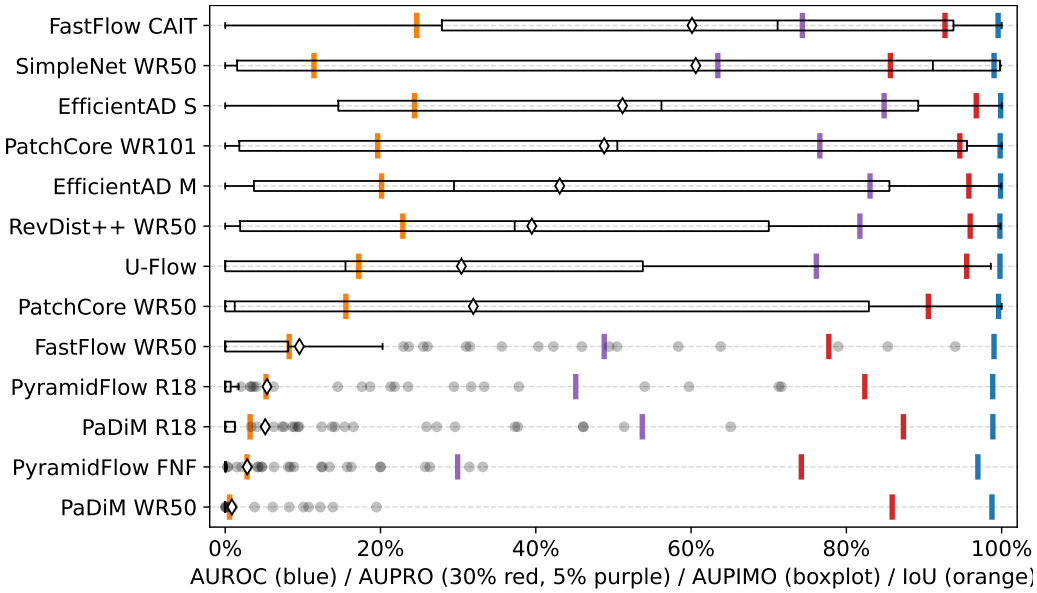


(c) Heatmaps. Images selected according to AUPIMO's statistics. Statistic and image index annotated on upper left corner.

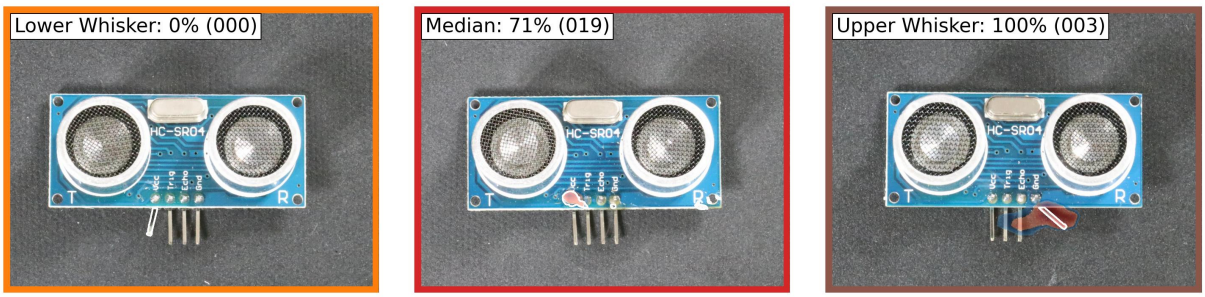
Figure A.25: Benchmark on VisA / Macaroni 2. PIMO curves and heatmaps are from EfficientAD M. 200 images (100 normal, 100 anomalous).



(a) Average rank diagram.

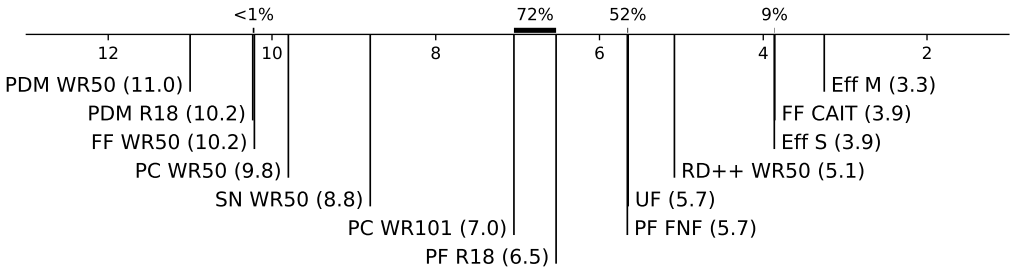


(b) Score distributions.

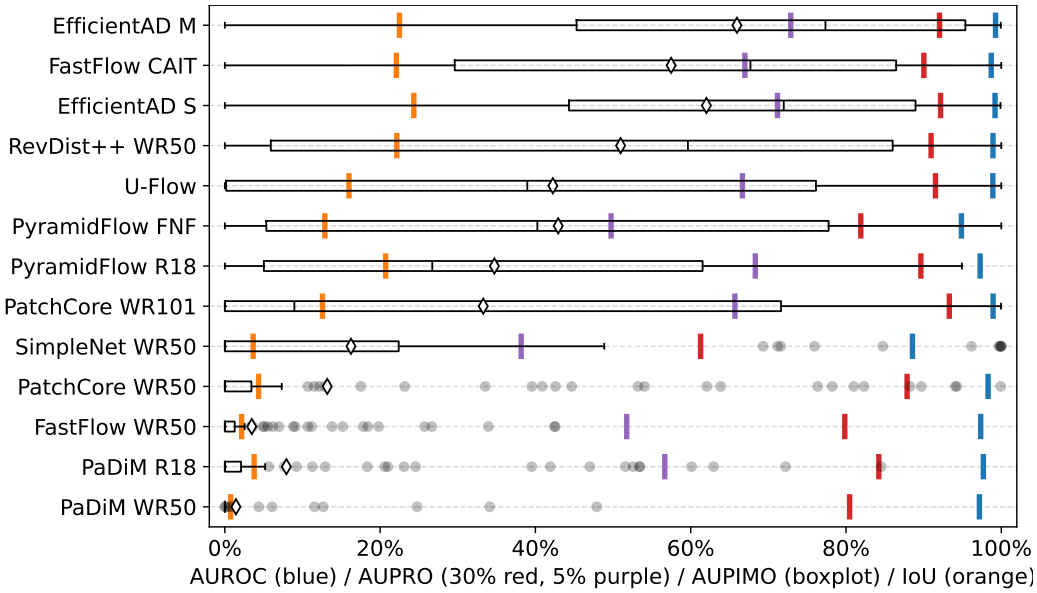


(c) Heatmaps. Images selected according to AUPIMO's statistics. Statistic and image index annotated on upper left corner.

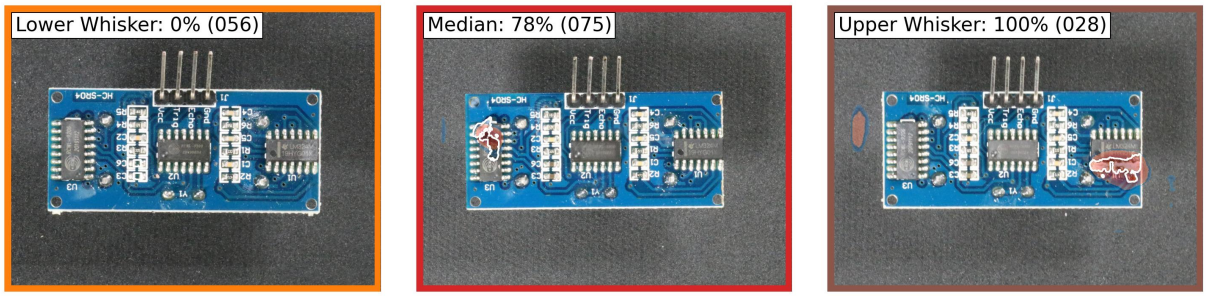
Figure A.26: Benchmark on VisA / PCB 1. PIMO curves and heatmaps are from FastFlow CAIT. 200 images (100 normal, 100 anomalous).



(a) Average rank diagram.

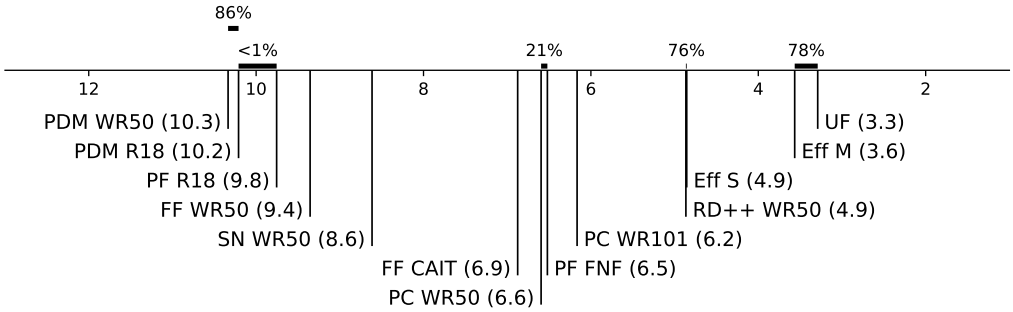


(b) Score distributions.

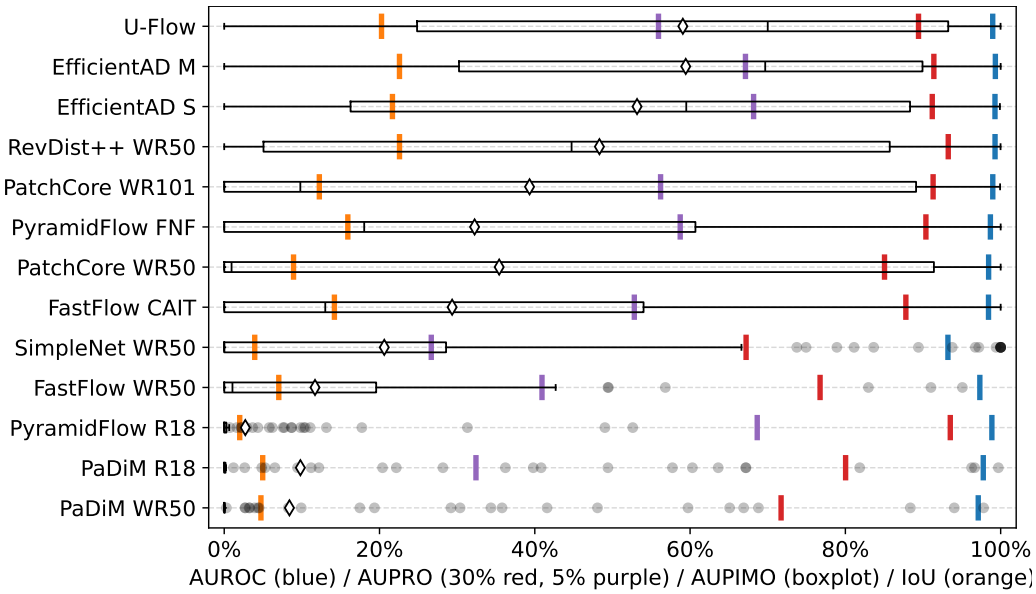


(c) Heatmaps. Images selected according to AUPIMO's statistics. Statistic and image index annotated on upper left corner.

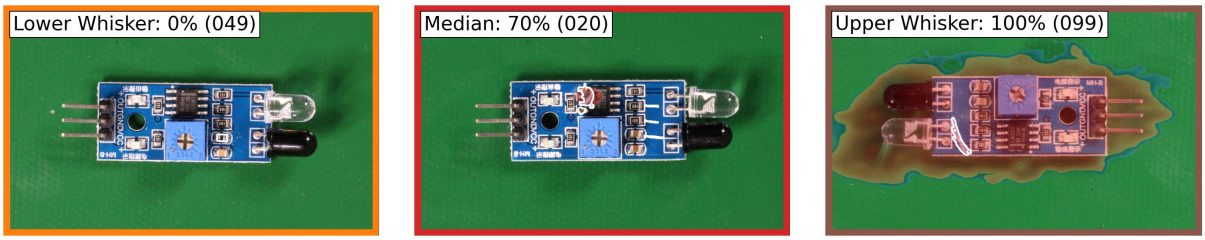
Figure A.27: Benchmark on VisA / PCB 2. PIMO curves and heatmaps are from EfficientAD M. 200 images (100 normal, 100 anomalous).



(a) Average rank diagram.

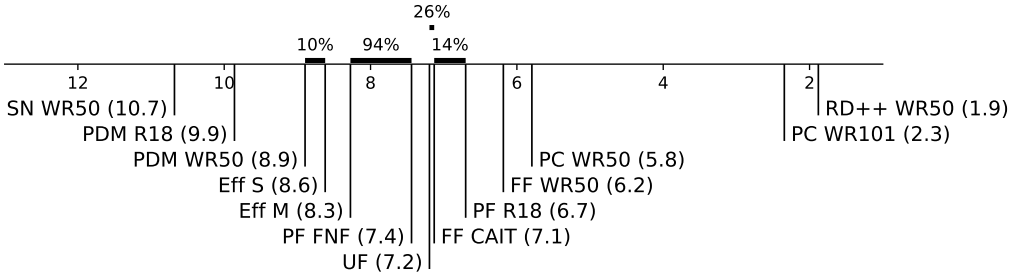


(b) Score distributions.

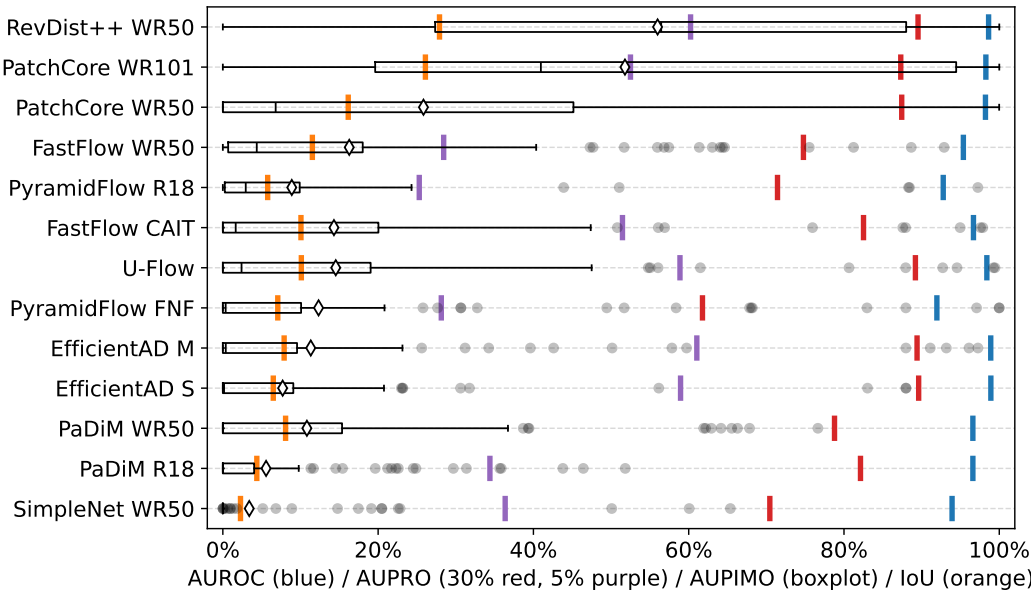


(c) Heatmaps. Images selected according to AUPIMO's statistics. Statistic and image index annotated on upper left corner.

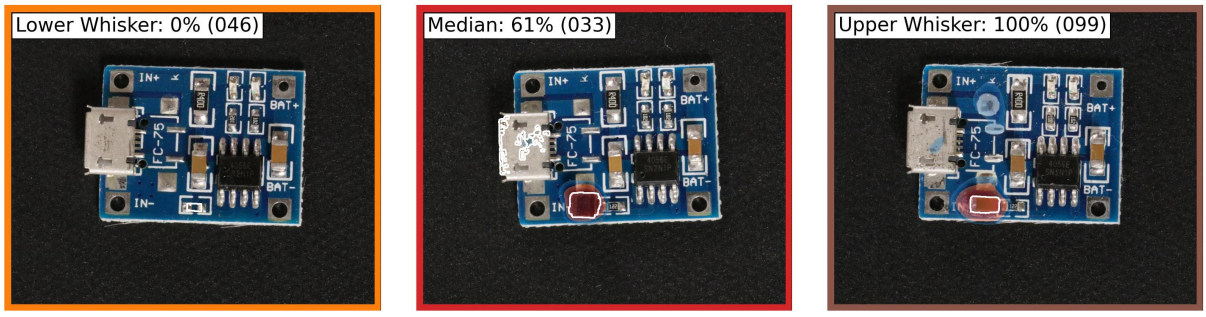
Figure A.28: Benchmark on VisA / PCB 3. PIMO curves and heatmaps are from U-Flow. 201 images (101 normal, 100 anomalous).



(a) Average rank diagram.

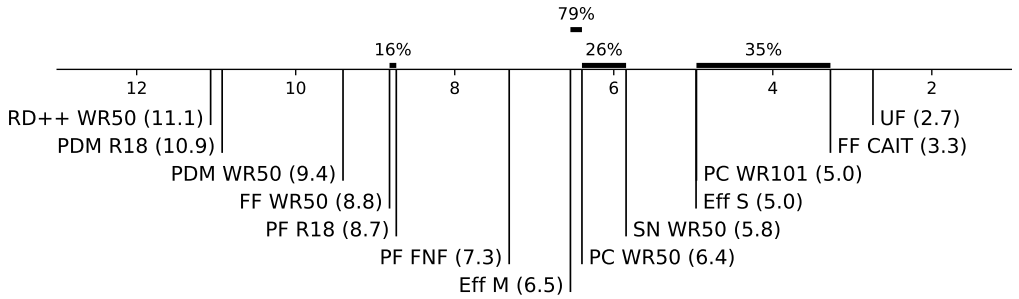


(b) Score distributions.

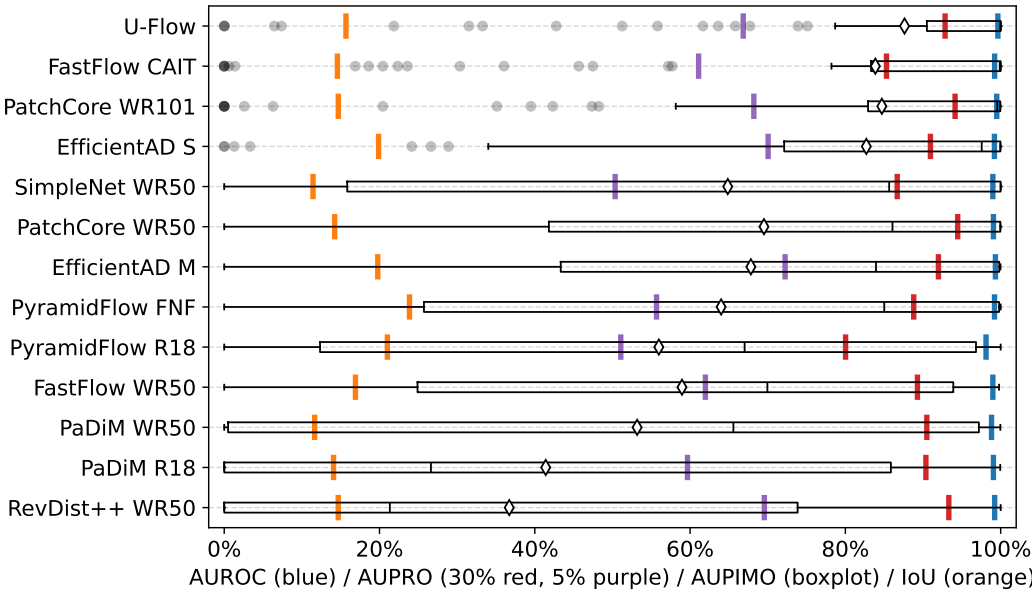


(c) Heatmaps. Images selected according to AUPIMO's statistics. Statistic and image index annotated on upper left corner.

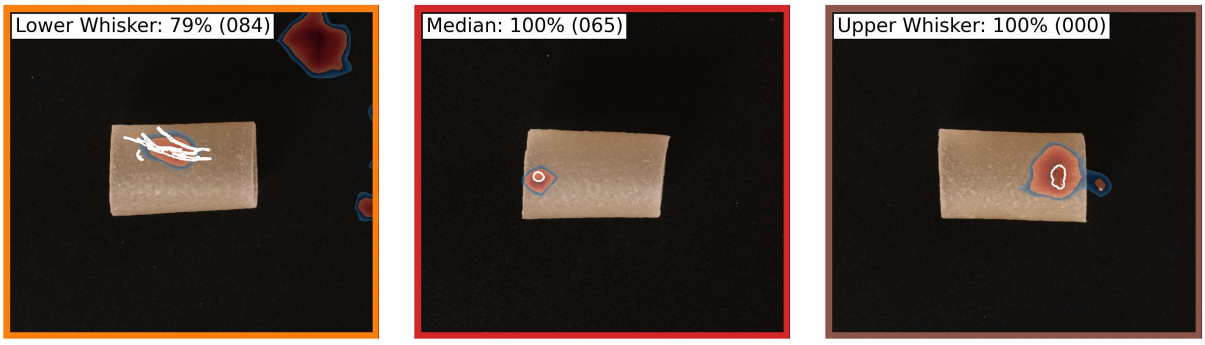
Figure A.29: Benchmark on VisA / PCB 4. PIMO curves and heatmaps are from RevDist++ WR50. 201 images (101 normal, 100 anomalous).



(a) Average rank diagram.



(b) Score distributions.

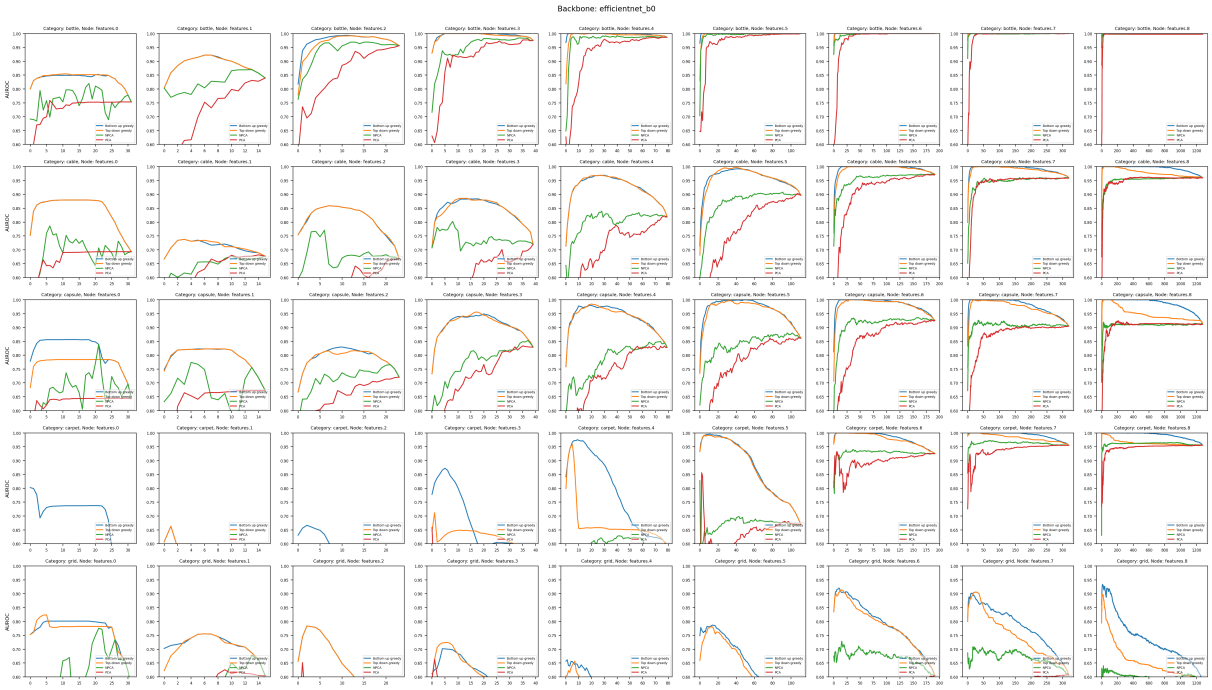


(c) Heatmaps. Images selected according to AUPIMO's statistics. Statistic and image index annotated on upper left corner.

Figure A.30: Benchmark on VisA / Pipe Fryum. PIMO curves and heatmaps are from U-Flow. 150 images (050 normal, 100 anomalous).

2 GreedyES Experiment 1

Fig. A.31 shows all the scenarios (categories and layers) from Experiment 1 as discussed in Section 2.2.2. The plots show the number of eigencomponents on the X-axis and the corresponding AUROC values on the Y-axis. It shows the AUROC performance of the greedy search for both traversal modes (bottom-up and top-down) compared to the results obtained from other dimension reduction strategies, including PCA, NPCA, and the supspace "$[\Phi_2, \Phi_3]$" introduced in [Lin, 2022].



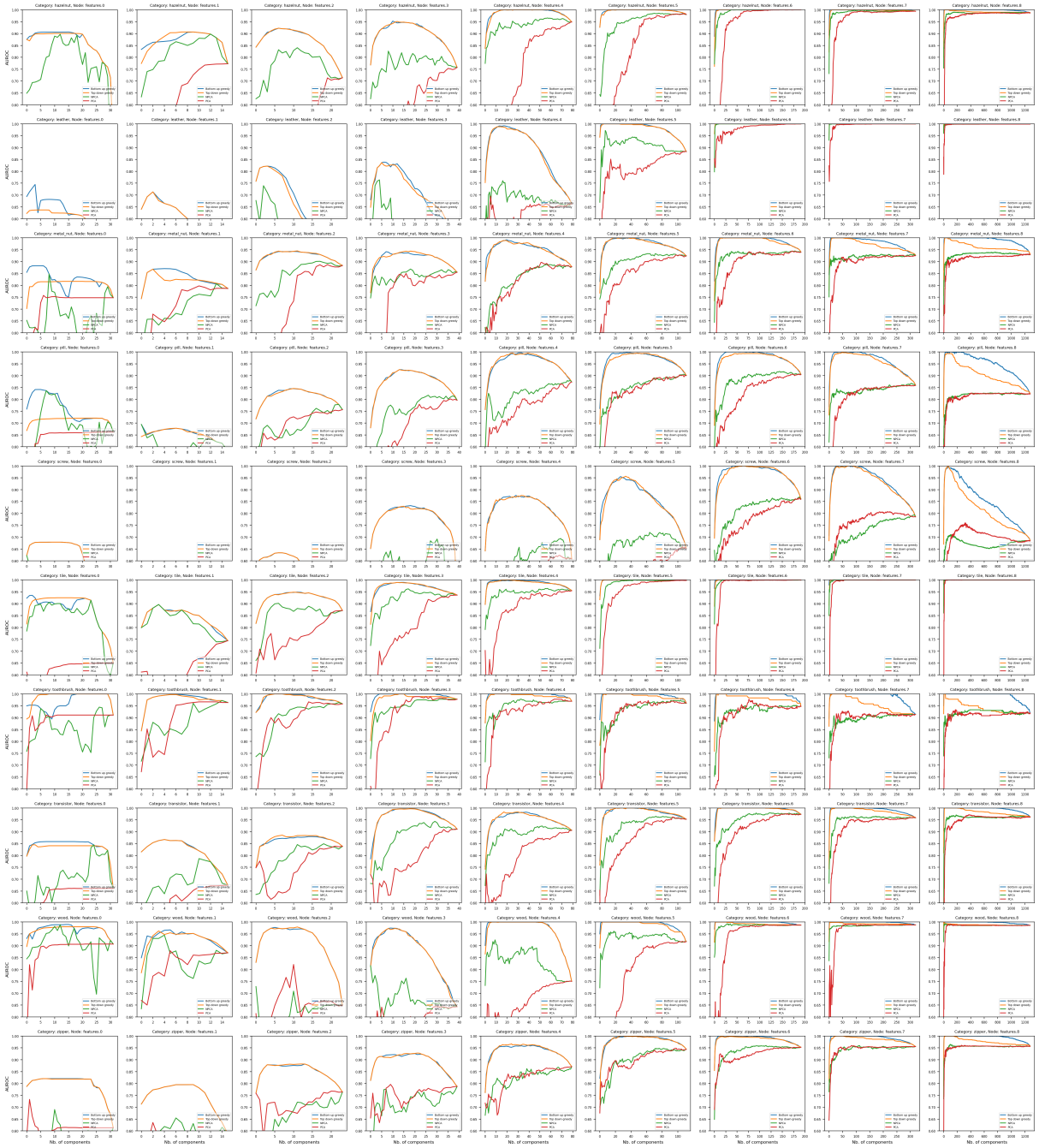
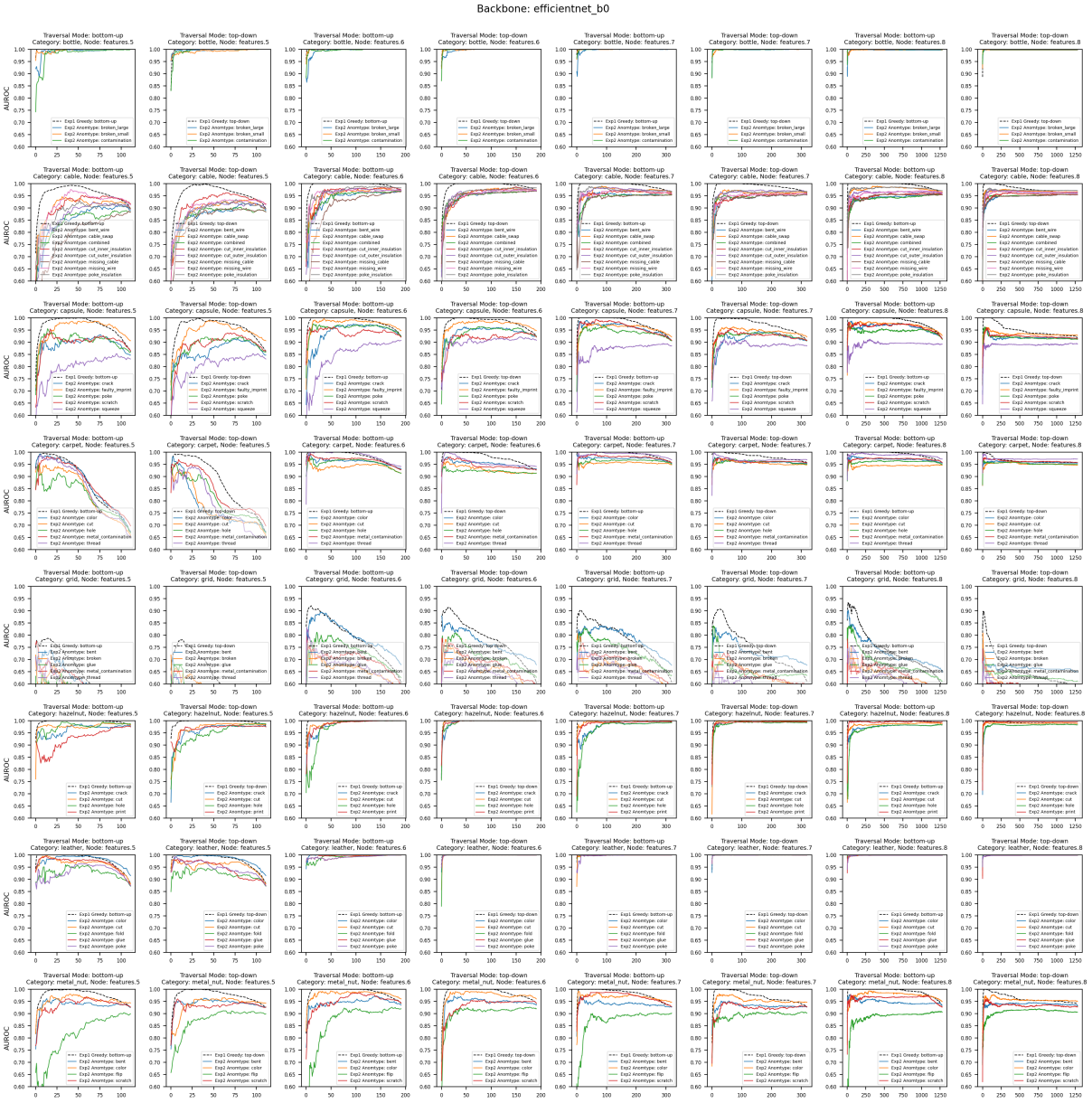


Figure A.31: Experiment 1: all plots.

3 GreedyES Experiment 2

Fig. A.32 shows all the scenarios (categories and layers) from Experiment 2 as discussed in Section 2.2.3. The plots show the number of eigencomponents on the X-axis and the corresponding AUROC values on the Y-axis. Each curve corresponds to a different $\mathcal{I}_{\text{opt}}$ / $\mathcal{I}_{\text{eval}}$ splits, so they do not match the same performance at the point $k = d$. The curve from Experiment 1 is also shown for reference. The category "toothbrush" is not in the results of this experiment as it only has one anomaly type.



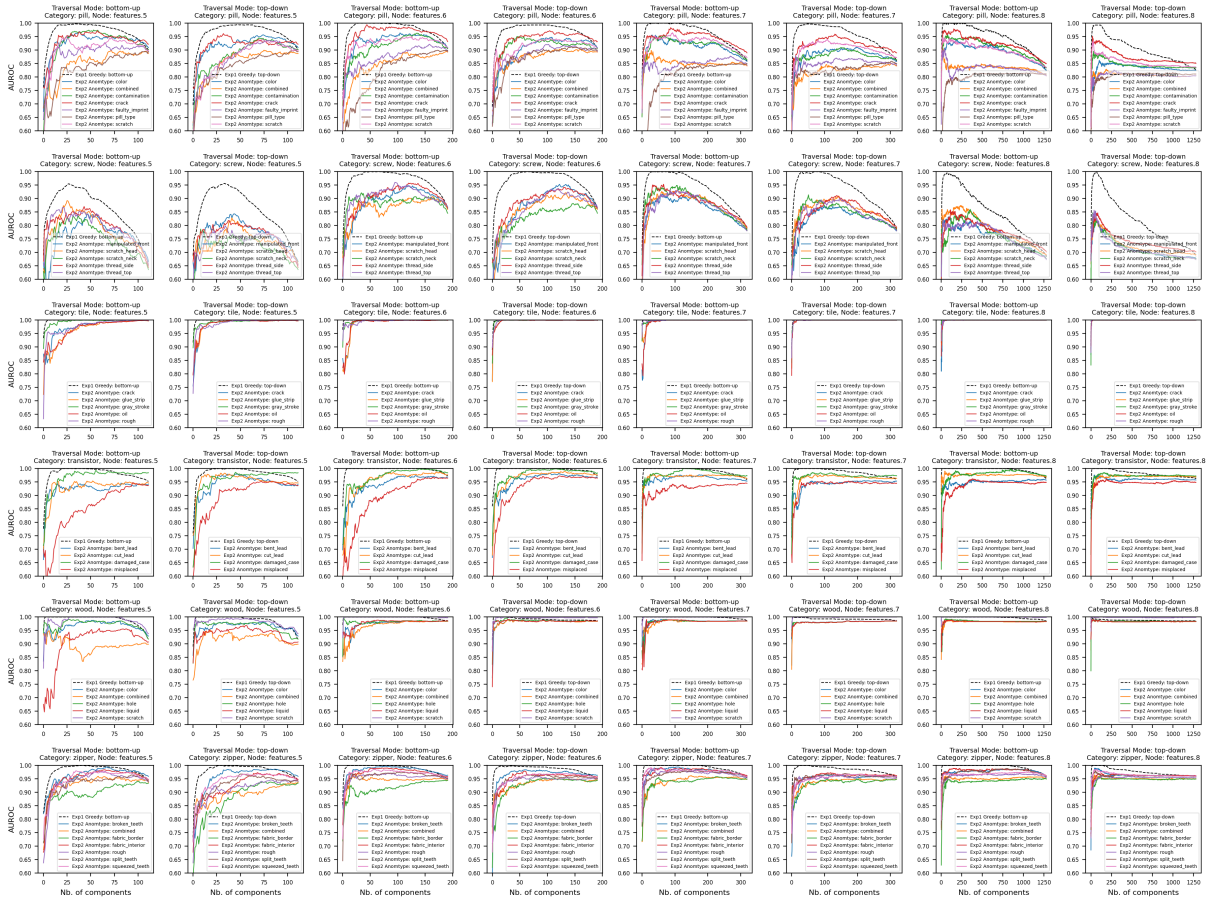
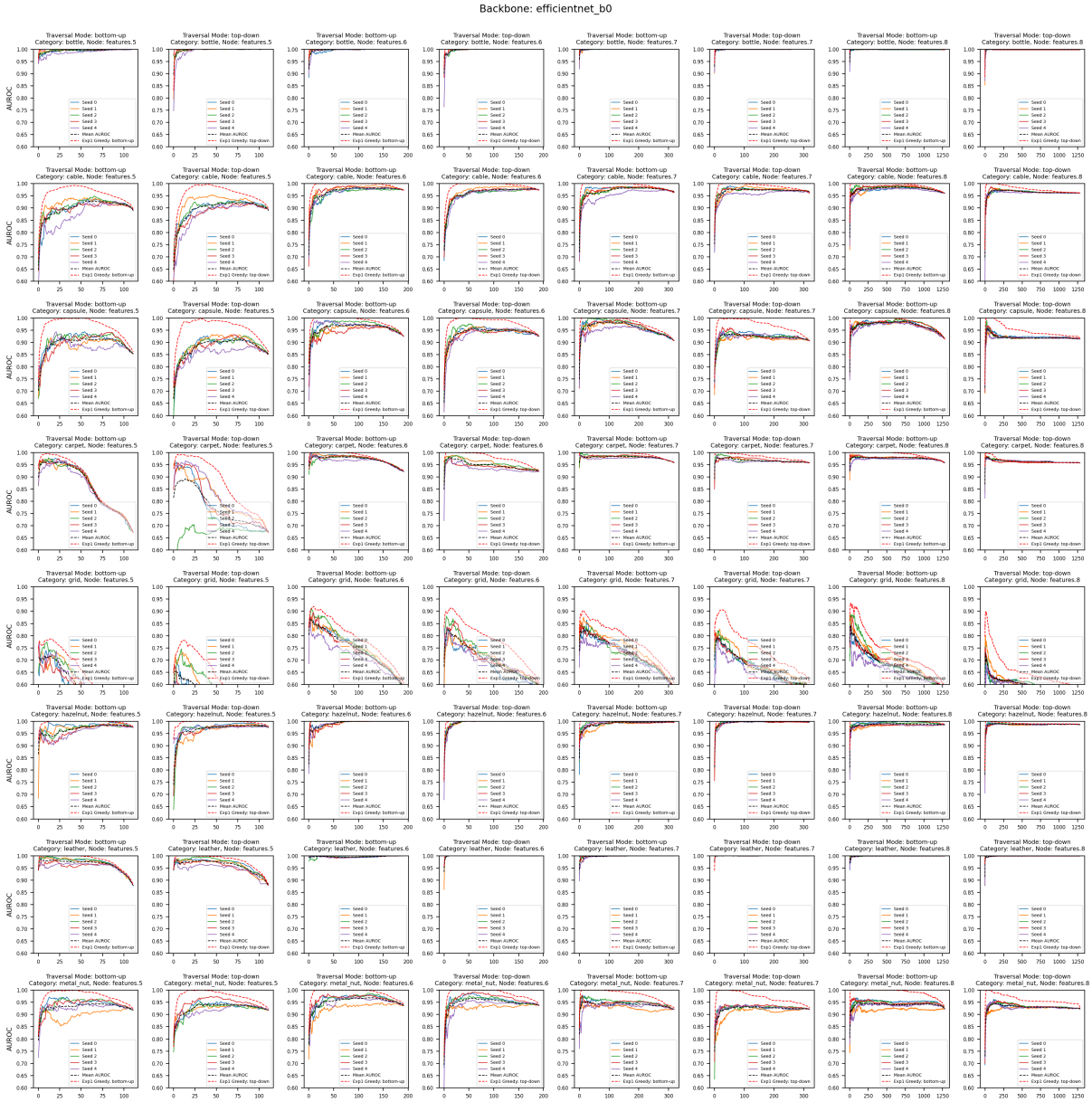


Figure A.32: Experiment 2: all plots.

4 GreedyES Experiment 3

Fig. A.33 shows all the scenarios (categories and layers) from Experiment 3 as discussed in Section 2.2.4. The plots show the number of eigencomponents on the X-axis and the corresponding AUROC values on the Y-axis. These plots show the performance of the various seeds and their cross-seed mean performance. The curve from Experiment 1 is also shown for reference.



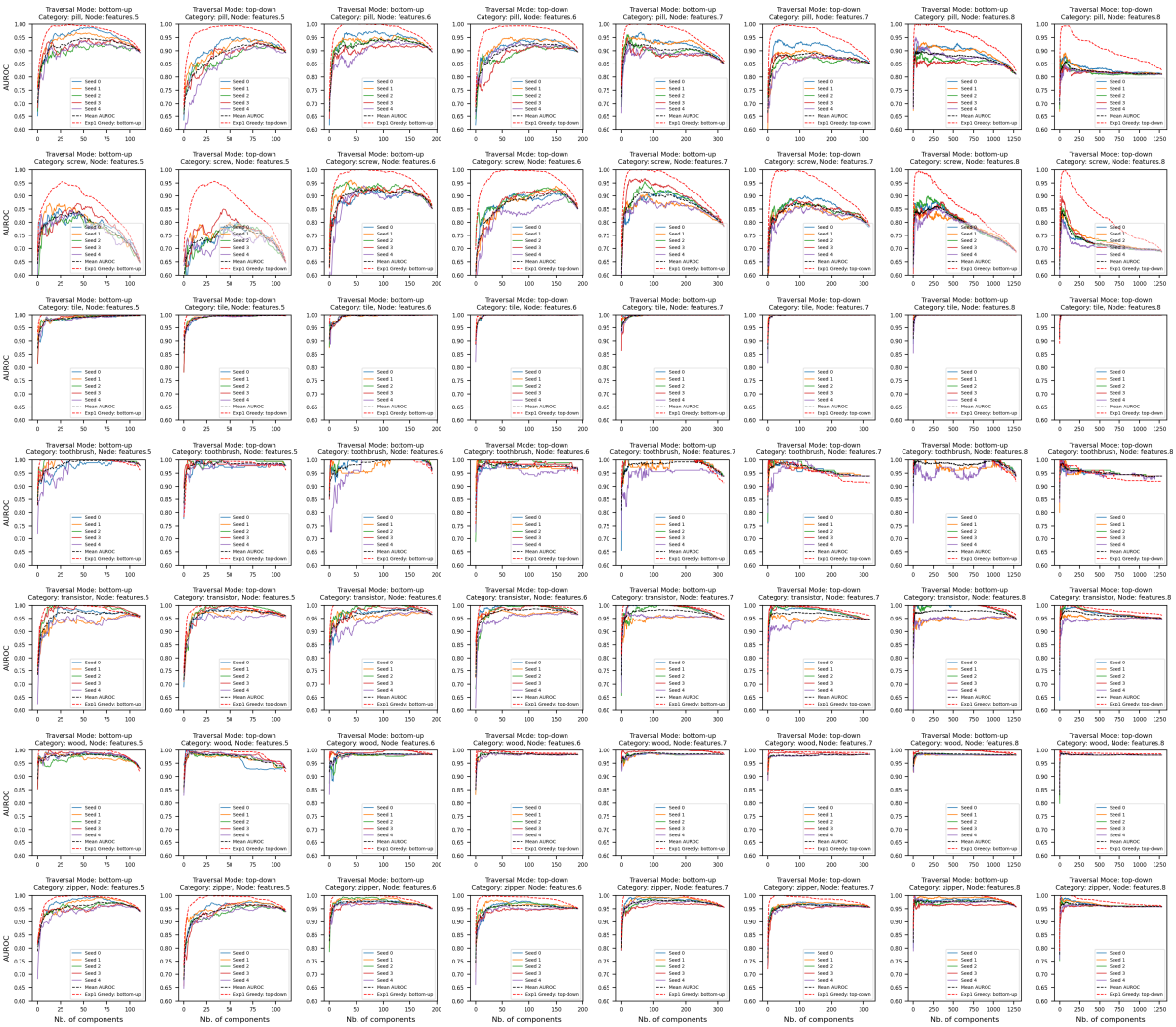


Figure A.33: Experiment 3: all plots.

RÉSUMÉ

Cette thèse traite de la détection d'anomalies visuelles, en particulier appliquée à l'identification non supervisée de défauts en applications d'inspection industrielle. Motivé par les récentes avancées dans le domaine, ce travail présente des contributions (1) à d'évaluation de la localisation des anomalies, (2) à la facilité d'utilisation des modèles via des techniques de post-traitement, et (3) à des améliorations spécifiques à des modèles basés sur les distributions gaussiennes. D'abord, nous présentons AUPIMO, une nouvelle mesure d'évaluation qui s'attaque aux limites des *benchmarks* actuels. Elle impose une minimisation stricte des faux positifs sur les échantillons normaux, encourageant ainsi une évaluation plus difficile. Des expériences menées sur 27 ensembles de données et huit modèles de l'état de l'art démontrent qu'AUPIMO fournit une évaluation plus fiable et plus détaillée des performances. Deuxièmement, nous proposons une méthode de sélection non supervisée de seuil et spécifique à l'image pour la localisation des anomalies. Cette méthode évite des biais introduits par les seuils basés sur des statistiques et offre une alternative aux *heatmaps* codées en couleur. Troisièmement, nous analysons les modèles basés sur distributions gaussiennes à l'aide d'une nouvelle méthode de réduction de la dimensionnalité. Cette analyse aboutit à des résultats qui remettent en question des notions dominantes dans le domaine, telles que la corrélation entre la variance et la performance en détection d'anomalies. Cette nouvelle méthode de réduction de la dimensionnalité basée sur les sous-espaces, combinée à des anomalies synthétiques, permet d'améliorer les performances de manière consistante. Nous présentons également un outil de visualisation permettant de localiser les anomalies pour les modèles gaussiens conçu pour des images entières. La thèse présente également une amélioration incrémentale d'un modèle de classification à une classe, permettant une utilisation plus efficace des annotations au niveau des pixels et un entrainement plus rapide. Les contributions proposées visent à combler l'écart entre la recherche et les applications du monde réel, en offrant des solutions agnostiques aux modèles pour améliorer l'analyse comparative et la facilité d'utilisation des modèles, ainsi que des améliorations spécifiques aux modèles basés sur les gaussiennes.

MOTS CLÉS

Détection d'Anomalies Visuelles, Apprentissage Non Supervisé, Métrique d'Évaluation, Localisation d'Anomalies, Modèles Gaussiens, Réduction de Dimensionnalité, *Benchmark*, Inspection Industrielle, AUPIMO

ABSTRACT

This thesis addresses visual anomaly detection, focusing on unsupervised defect identification in industrial inspection applications. Motivated by recent advances in the field, this work presents contributions to (1) the evaluation standards of anomaly localization, (2) usability of models via post-processing techniques, and (3) model-specific improvements. First, we introduce AUPIMO, a novel evaluation metric that addresses limitations of existing benchmarks. It imposes a hard minimization of false positives on normal samples, encouraging a more challenging and trustworthy evaluation. Experiments across 27 datasets and eight state-of-the-art models demonstrate that AUPIMO provides a more reliable and detailed performance assessment than previous benchmarks. Second, we propose an unsupervised, image-specific threshold selection method for anomaly localization. This method avoids biases introduced by statistics-based thresholds, offering an alternative to color-coded heatmaps. Third, we analyze Gaussian distribution-based models through a novel dimensionality reduction method. This analysis leads to findings that challenge prevailing notions in the field, such as the correlation between variance and model performance. This novel subspace-based dimensionality reduction method, combined with synthetic anomalies, is shown to consistently improve performance. Additionally, we introduce a visualization tool enabling anomaly localization for image-wise Gaussian models. The thesis also presents an incremental improvement to a pixel-wise one-class classification model, enabling more effective use of pixel-wise annotations and faster training. The proposed contributions aim to bridge the gap between research and real-world applications, offering model-agnostic solutions to improve benchmarking and model usability, and model-specific improvements to Gaussian-based models.

KEYWORDS

Visual Anomaly Detection, Unsupervised Learning, Evaluation Metric, Anomaly Localization, Gaussian Models, Dimensionality Reduction, Benchmark, Industrial Inspection, AUPIMO