International Journal of Computer Vision (2021) 129:1038–1059
https://doi.org/10.1007/s11263-020-01400-4

The MVTec Anomaly Detection Dataset: A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection

Paul Bergmann[1,2] · **Kilian Batzner**[1] · **Michael Fauser**[1] · **David Sattlegger**[1] · **Carsten Steger**[1]

Received: 15 April 2020 / Accepted: 2 November 2020 / Published online: 6 January 2021

Abstract
The detection of anomalous structures in natural image data is of utmost importance for numerous tasks in the field of computer vision. The development of methods for unsupervised anomaly detection requires data on which to train and evaluate new approaches and ideas. We introduce the MVTec anomaly detection dataset containing 5354 high-resolution color images of different object and texture categories. It contains normal, i.e., defect-free images intended for training and images with anomalies intended for testing. The anomalies manifest themselves in the form of over 70 different types of defects such as scratches, dents, contaminations, and various structural changes. In addition, we provide pixel-precise ground truth annotations for all anomalies. We conduct a thorough evaluation of current state-of-the-art unsupervised anomaly detection methods based on deep architectures such as convolutional autoencoders, generative adversarial networks, and feature descriptors using pretrained convolutional neural networks, as well as classical computer vision methods. We highlight the advantages and disadvantages of multiple performance metrics as well as threshold estimation techniques. This benchmark indicates that methods that leverage descriptors of pretrained networks outperform all other approaches and deep-learning-based generative models show considerable room for improvement.

Keywords Anomaly detection · Novelty detection · Datasets · Unsupervised learning · Defect segmentation

1 Introduction

Humans are very good at recognizing whether an image is similar to what they have previously observed or whether it is something novel or anomalous. So far, machine learning systems, however, seem to have difficulties with such tasks.

There are many relevant applications that must rely on unsupervised algorithms that can detect anomalous regions. In the manufacturing industry, for example, optical inspection tasks often lack defective samples that could be used for supervised training or it is unclear which kinds of defects may appear. In active learning systems, structures that are identified as anomalous might indicate the necessity of including a specific image for training. Therefore, it is not surprising that recently a significant amount of interest has been directed towards anomaly detection in natural image data using modern machine learning architectures. Several terms are used more or less equivalently in the literature to describe such types of problem settings, such as anomaly detection, novelty detection, outlier detection, or one-class classification. We would like to differentiate between the following two complementary problem settings. In this work, *novelty detection* refers to image-level classification settings in which the inlier and outlier distributions differ significantly. *Anomaly detection*, on the other hand, shall be defined as the task of finding and ideally segmenting anomalies in images that are

Communicated by Daniel Scharstein.

✉ Paul Bergmann
paul.bergmann@mvtec.com
https://www.mvtec.com

Kilian Batzner
kilian.batzner@mvtec.com

Michael Fauser
fauser@mvtec.com

David Sattlegger
sattlegger@mvtec.com

Carsten Steger
steger@mvtec.com

[1] MVTec Software GmbH, Arnulfstr. 205, 80634 Munich, Germany

[2] Department of Informatics, Technical University of Munich, Boltzmannstraße 3, 85748 Garching, Germany

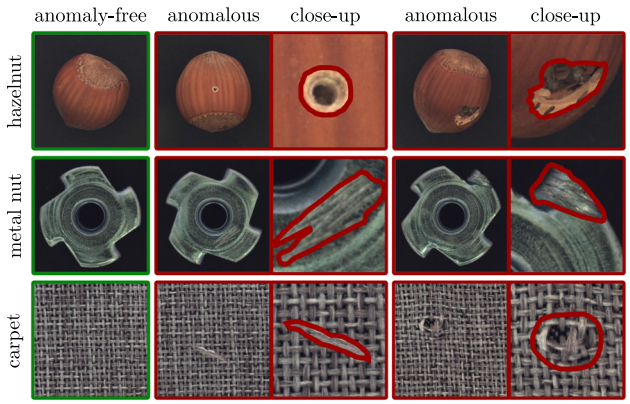

Fig. 1 Two objects (*hazelnut* and *metal nut*) and one texture (*carpet*) from the MVTec Anomaly Detection dataset. For each of them, one defect-free image and two images that contain anomalies are displayed. Anomalous regions are highlighted in close-up figures together with their pixel-precise ground truth labels. The dataset contains objects and textures from several domains and covers various anomalies that differ in attributes such as size, color, and structure

very close to the training data, i.e., differ only in subtle deviations in possibly very small, confined regions.

A number of algorithms have been proposed that test whether a network is able to detect whether new input data matches the distribution of the training data. Many of them, however, focus on image-level novelty detection, for which an established evaluation protocol is to arbitrarily label a number of classes from existing object classification datasets as outlier classes and use the remaining classes as inliers for training. It is then measured how well the trained algorithm can distinguish between previously unseen outlier and inlier samples. While the detection of outliers on an image level is important and has received much attention from the research community, only a small amount of work has been directed towards solving anomaly detection problems. To encourage the development of machine learning models to tackle this problem and evaluate their performance, we require suitable data. Curiously, there is a lack of comprehensive real-world datasets for anomaly detection.

In many areas of computer vision, large-scale datasets have led to incredible advances during the last few years. Consider how closely intertwined the development of new classification methods is with the introduction of datasets such as MNIST (LeCun et al. 1998), CIFAR10 (Krizhevsky and Hinton 2009), or ImageNet (Krizhevsky et al. 2012).

To the best of our knowledge, no comprehensive large-scale, high-resolution dataset exists for the task of unsupervised anomaly detection. As a first step to fill this gap and to spark further research in the development of new methods in this area, we introduce the MVTec Anomaly Detection (MVTec AD or MAD for short) dataset[1] that facilitates a thorough evaluation of such methods. Some example images are shown in Fig. 1. We identify industrial inspection tasks as an ideal and challenging real-world use case for these scenarios. Defect-free images of objects or textures are used to train a model that must determine whether an anomaly is present during test time. Unsupervised methods play a significant role here since it is often unknown beforehand which types of defects might occur during manufacturing. In addition, industrial processes are highly optimized in order to minimize the number of defective samples. Therefore, only a very limited amount of images with defects is available, in contrast to a vast amount of defect-free samples that can be used for training. Ideally, methods should provide a pixel-accurate segmentation of anomalous regions. All this makes industrial inspection tasks perfect benchmarks for unsupervised anomaly detection methods that work on natural images.

The present work is an extension of Bergmann et al. (2019a), which presents the MVTec AD dataset together with an initial baseline evaluation of state-of-the-art deep learning and traditional anomaly detection models. This paper adds a thorough evaluation that employs multiple performance measures and discusses their advantages and shortcomings in the context of anomaly detection. Furthermore, we introduce techniques for selecting thresholds that allow to obtain binary anomaly predictions. We benchmark updated implementations of the methods considered in the preceding version of this paper and include an additional recent deep learning-based approach (Bergmann et al. 2020). We further extend the evaluation by providing a discussion of the execution time and memory consumption of the evaluated methods. Overall, our main contributions in this extended work are:

- We introduce a novel and comprehensive dataset for the task of unsupervised anomaly detection in natural image data. It mimics real-world industrial inspection scenarios and consists of 5354 high-resolution images of five unique textures and ten unique objects from different domains. There are 73 different types of anomalies in the form of defects or structural deviations in the objects or textures. For each defect image, we provide pixel-accurate ground truth regions (1888 in total) that allow to evaluate methods for both anomaly classification and segmentation.
- We conduct a thorough evaluation of current state-of-the-art methods as well as more traditional methods for unsupervised anomaly segmentation on the dataset. We show that the evaluated methods do not perform equally well across object and defect categories. Methods that leverage descriptors of pretrained networks outperform

[1] The entire dataset is publicly available for download at https://www.mvtec.com/company/research/datasets.

all other evaluated approaches. Generative deep learning methods that are trained from scratch show considerable room for improvement.
– We provide a thorough discussion on various evaluation metrics and threshold estimation techniques for unsupervised anomaly segmentation and highlight their advantages and shortcomings. Our evaluations demonstrate the importance of selecting suitable metrics and show that threshold selection is a highly challenging task in practice. In addition, we include a discussion about the runtime and memory consumption of the evaluated methods. These are important criteria for the applicability of the benchmarked methods in real-world scenarios such as automated inspection tasks.

2 Related Work

2.1 Existing Datasets for Anomaly Detection

We first give a brief overview of datasets that are commonly used for anomaly detection in natural images and demonstrate the need for our novel dataset. We distinguish between datasets that are designed for a simple binary decision between anomalous and anomaly-free images and datasets that allow for the segmentation of anomalous regions.

2.1.1 Classification of Anomalous Images

When evaluating methods for outlier detection in multi-class classification scenarios, a common practice is to adapt existing classification datasets for which class labels are already available. The most prominent examples are MNIST (LeCun et al. 1998), CIFAR10 (Krizhevsky and Hinton 2009), and ImageNet (Krizhevsky et al. 2012). A popular approach is to select an arbitrary subset of classes, re-label them as outliers, and train a novelty detection system solely on the remaining inlier classes (An and Cho 2015; Chalapathy et al. 2018; Ruff et al. 2018; Burlina et al. 2019). During the testing phase, it is checked whether the trained model is able to correctly predict that a test sample belongs to one of the inlier classes. An alternative approach is to train a classifier on all classes of a single dataset, e.g., MNIST, and use images of an entirely different dataset, e.g., notMNIST (Bulatov 2011), as outliers. While these approaches immediately provide a large amount of data for training and testing, the anomalous samples differ significantly from the samples drawn from the training distribution. Therefore, when performing evaluations on such datasets, it is unclear how a proposed method would generalize to data where anomalies manifest themselves in less significant deviations from the training data manifold.

For this purpose, Saleh et al. (2013) propose a dataset that contains six categories of abnormally shaped objects, such as oddly shaped cars, airplanes, and boats, obtained from internet search engines, that should be distinguished from regular samples of the same class in the PASCAL VOC dataset (Everingham et al. 2015). While this data might be closer to the training data manifold, the decision is again based on entire images rather than finding the parts of the images that make them novel or anomalous.

2.1.2 Segmentation of Anomalous Regions

For the evaluation of methods that segment anomalies in images, only very few datasets are currently available to the public. Many of them are limited to the inspection of textured surfaces or focus on novelty detection in multi-class semantic segmentation scenarios. To the best of our knowledge, there does not yet exist a comprehensive dataset that allows for the segmentation of anomalous regions in natural images where the anomalies manifest themselves in subtle deviations from the training data.

Carrera et al. (2017) provide NanoTWICE,[2] a dataset of 45 gray-scale images that show a nanofibrous material acquired by a scanning electron microscope. Five defect-free images can be used for training. The remaining 40 images contain anomalous regions in the form of specks of dust or flattened areas. Since the dataset only provides a single kind of texture, it is unclear how well algorithms that are evaluated on this dataset generalize to other textures of different domains.

A dataset that is specifically designed for optical inspection of textured surfaces was proposed by Wieler and Hahn (2007). They provide ten classes of artificially generated gray-scale textures with defects weakly annotated in the form of ellipses. Each class comprises 1000 defect-free texture patches for training and 150 defective patches for testing. The annotations, however, are coarse and since the textures were generated by very similar texture models, the variance in appearance between the different textures is insignificant. Furthermore, artificially generated datasets can only be seen as an approximation to the real world.

Huang et al. (2018) introduce a surface inspection dataset of magnetic tiles. It contains 1344 grayscale images of a single texture. Each image is either anomaly-free or contains one of five different surface defects, such as cracks or uneven areas. For each defective image, pixel-precise ground-truth labels are provided. Similarly, Song and Yan (2013) introduce a database of 1800 grayscale images of a single steel surface. It contains six different defect types, such as scratches or surface crazings. Each defect is coarsely annotated with a bounding box.

Blum et al. (2019) recently introduced Fishyscapes, a dataset intended to benchmark semantic segmentation algorithms with respect to their ability to detect out-of-

[2] http://www.mi.imati.cnr.it/ettore/NanoTWICE/.

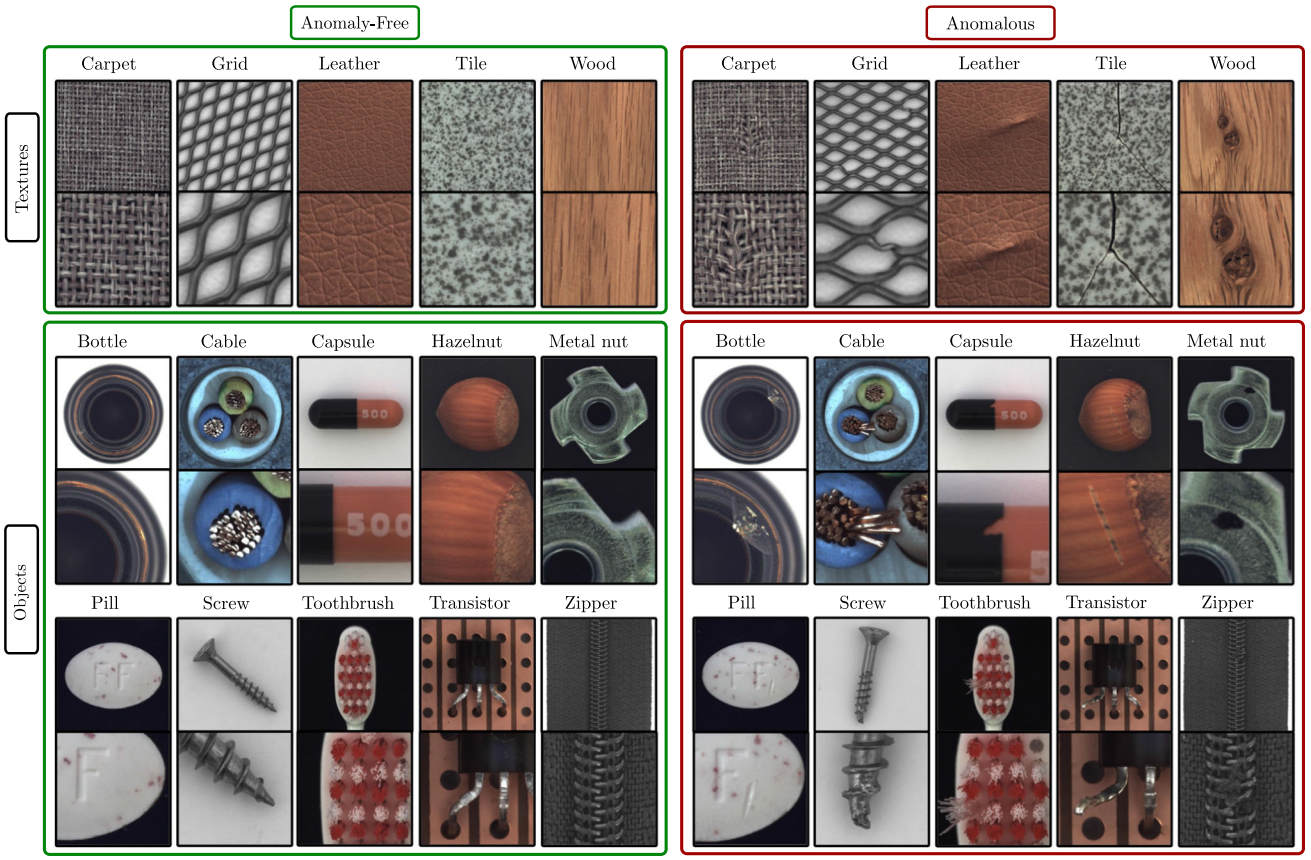

Fig. 2 Example images for all five textures and ten object categories of the MVTec Anomaly Detection dataset. For each category, an anomaly-free as well as an anomalous example is shown. The top row shows the entire input image. The bottom row gives a close-up view. For anomalous images, the close-up highlights the anomalous regions

distribution inputs. They artificially inserted images of novel objects into images of the Cityscapes dataset (Cordts et al. 2016), for which pixel-precise annotations are available. The task is then to train a model for semantic segmentation while at the same time being able to identify certain objects as novelties by leveraging the model's per-pixel uncertainties. In contrast to their dataset, we focus on the one-class setting, where dataset images only show a single object and no training annotations are available. Furthermore, our anomalies manifest themselves in subtle deviations from the input images rather than showing entirely different object classes.

The CAOS (Combined Anomalous Object Segmentation) benchmark introduced by Hendrycks et al. (2019a) provides two datasets similar to Fishyscapes. It consists of the StreetHazards and BDD-Anomaly datasets. StreetHazards contains artificially rendered driving scenes with inserted foreign objects. BDD-Anomaly also consists of driving scenes and was derived from the BDD100K dataset (Yu et al. 2020) by selecting two classes as anomalous and removing images containing these classes from the training and validation sets. As in the case of Fishyscapes, the CAOS datasets are geared towards a multi-class setting.

2.2 Methods

The landscape of methods for unsupervised anomaly detection is diverse and many approaches have been suggested to tackle the problem (An and Cho 2015; Pereraet and Patel 219). Pimentel et al. (2014) and Ehret et al. (2019) give a comprehensive review of existing work. We restrict ourselves to a brief overview of current state-of-the-art methods that are able to segment anomalies, focusing on those that serve as baselines for our benchmark on the dataset.

2.2.1 Generative Adversarial Networks (GANs)

Schlegl et al. (2017) propose to model the training data manifold by a generative adversarial network (Goodfellow et al. 2014) that is trained solely on defect-free images. The generator is able to produce images that fool a simultaneously trained discriminator network in an adversarial way.

Table 1 Statistical overview of the MVTec AD dataset

	Category	# Train	# Test (good)	# Test (defective)	# Defect groups	# Defect regions	Image side length	Grayscale
Textures	Carpet	280	28	89	5	97	1024	
	Grid	264	21	57	5	170	1024	✓
	Leather	245	32	92	5	99	1024	
	Tile	230	33	84	5	86	840	
	Wood	247	19	60	5	168	1024	
Objects	Bottle	209	20	63	3	68	900	
	Cable	224	58	92	8	151	1024	
	Capsule	219	23	109	5	114	1000	
	Hazelnut	391	40	70	4	136	1024	
	Metal nut	220	22	93	4	132	700	
	Pill	267	26	141	7	245	800	
	Screw	320	41	119	5	135	1024	✓
	Toothbrush	60	12	30	1	66	1024	
	Transistor	213	60	40	4	44	1024	
	Zipper	240	32	119	7	177	1024	✓
	Total	3629	467	1258	73	1888	-	-

For each category, the number of training and test images is given together with additional information about the defects present in the respective test images

For anomaly detection, the algorithm searches for a latent sample that reproduces a given input image and manages to fool the discriminator. Anomaly maps can be obtained by a pixelwise comparison of the reconstructed image with the original input.

The search for a suitable latent sample requires the solution of a nonlinear optimization problem during inference, e.g., by means of gradient descent. This makes their approach computationally expensive. For faster inference, Schlegl et al. (2019) propose to train an additional encoder network that maps input images to their respective latent samples with a single forward pass.

Instead of comparing each pixel of the input image with the one in the resynthesized image directly, Lis et al. (2019) propose to train a discrepancy network on artificially generated anomalies that directly outputs the regions where the reconstruction failed. Since their method requires pixel-precise semantic annotations of the training data, we do not consider this method for our benchmark.

2.2.2 Deep Convolutional Autoencoders

Convolutional Autoencoders (CAEs) (Goodfellow et al. 2016) are commonly used as a base architecture in unsupervised anomaly detection settings. They attempt to reconstruct defect-free training samples through a bottleneck (latent space). During testing, they should be unable to reproduce images that differ from the data that was observed during training. Anomalies are detected by a per-pixel comparison of the input with its reconstruction. Recently, Bergmann et al. (2019b) pointed out the disadvantages of per-pixel loss functions in autoencoding frameworks when used in anomaly segmentation scenarios and proposed to incorporate spatial information of local patch regions using structural similarity (Wang et al. 2004) for improved segmentation results.

There exist various extensions to CAEs such as memory-augmented (Gong et al. 2019) or variational autoencoders (VAEs) (Kingma and Welling 2014). The latter have been used by Baur et al. (2019) for the unsupervised segmentation of anomalies in brain MR scans. They, however, do not report significant improvements over the use of standard CAEs. This coincides with the observations made by Bergmann et al. (2019b). Nalisnick et al. (2019) and Hendrycks et al. (2019b) provide further evidence that probabilities obtained from VAEs and other deep generative models might fail to model the true likelihood of the training data. Therefore, we restrict ourselves to deterministic autoencoder frameworks in the evaluation of various methods on our dataset in Sect. 6.

2.2.3 Features of Pretrained Convolutional Neural Networks

The aforementioned approaches attempt to learn feature representations solely from the provided training data. In addition, there are several methods that use feature descriptors obtained from CNNs that have been pretrained on a separate image classification task.

Napoletano et al. (2018) propose to use clustered feature descriptions obtained from the activations of a ResNet-18 (He et al. 2016) classification network pretrained on ImageNet to distinguish normal from anomalous data. We refer to their method as CNN Feature Dictionary. Training features are extracted from patches that are cropped at random locations from the input images and their distribution is modeled with a K-Means classifier. Since feature extraction for all possible image patches quickly becomes prohibitively expensive and the capacity of K-Means classifiers is limited, the total number of available training features is typically heavily subsampled. This method achieves state-of-the-art results on the NanoTWICE dataset. Being designed for one-class classification, it only provides a binary decision whether an input image contains an anomaly or not. In order to obtain a spatial anomaly map, the classifier must be evaluated at multiple image locations, ideally at each single pixel. This quickly becomes a performance bottleneck for large images. To increase performance in practice, not every pixel location is evaluated and the resulting anomaly maps are therefore coarse.

Sabokrou et al. (2018) model the distribution of descriptors extracted from the first layers of an AlexNet pretrained on ImageNet with a unimodal Gaussian distribution. The fully convolutional architecture of the employed network allows for efficient feature extraction during training and inference. However, the use of pooling layers rapidly downsamples the input image and leads to a loss in resolution of the output anomaly map. Furthermore, unimodal Gaussian distributions cannot capture highly complex feature distributions. To tackle this issue, Marchal et al. (2020) propose to model the feature distribution with deep normalizing flows. They show that using a model with increased capacity indeed improves the performance over shallow distribution models.

Bergmann et al. (2020) propose a student–teacher framework that also leverages networks pretrained on ImageNet for the unsupervised segmentation of anomalous regions. An ensemble of randomly initialized student networks is trained to regress descriptors of pretrained teacher networks on anomaly-free data. During inference, the student networks fail to correctly predict the teachers' descriptors for anomalous regions and yield increased regression errors as well as predictive uncertainties. In contrast to the CNN Feature Dictionary, which requires heavy training data subsampling, this approach is trained on all available feature vectors. Since student and teacher networks densely extract local descriptors for each image pixel with a single forward pass, dense anomaly scores for each image pixel can be obtained with a single forward pass through each student and teacher network.

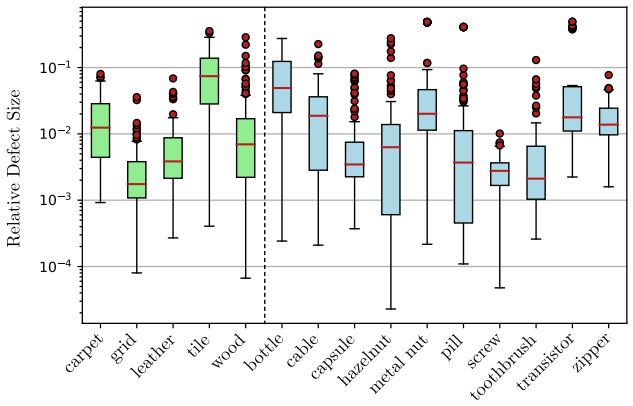

Fig. 3 Size of anomalies for all textures (green) and objects (blue) in the dataset on a logarithmic scale visualized as a box-and-whisker plot with outliers. Defect areas are reported as the number of pixels within a connected component relative to the total number of pixels within an image. Anomalies vary greatly in size for each dataset category

2.2.4 Traditional Methods

We also consider two traditional methods for our benchmark. Böttger and Ulrich (2016) extract hand-crafted feature descriptors from defect-free texture images. The distribution of feature vectors is modeled by a Gaussian Mixture Model (GMM) and anomalies are detected for extracted feature descriptors for which the GMM yields a low probability. Their algorithm was originally intended to be applied to images of regular textures. Nevertheless, it can also be applied to the objects of our dataset.

The second method is called Variation Model (Steger et al. 2018, Chapter 3.4.1.4). This method requires the objects in question to be aligned. Then, a reference image is constructed by calculating the pixelwise mean over a set of training images. In order for small perturbations of the object's shape to be tolerated, one defines permissible variations by calculating the standard deviation of the gray values of each pixel. For multichannel images, one can simply do this separately for each channel. During inference, a statistical test is performed for each image pixel that measures the deviation of the pixel's gray value from the reference. This deviation is used to construct an anomaly map.

3 Description of the Dataset

The MVTec Anomaly Detection dataset comprises 15 categories with 3629 images for training and 1725 images for testing. The training set contains only images without defects. The test set contains both: images containing various types of defects and defect-free images. Table 1 gives an overview for each object category. Some example images for every category together with an example defect are shown in Fig. 2.

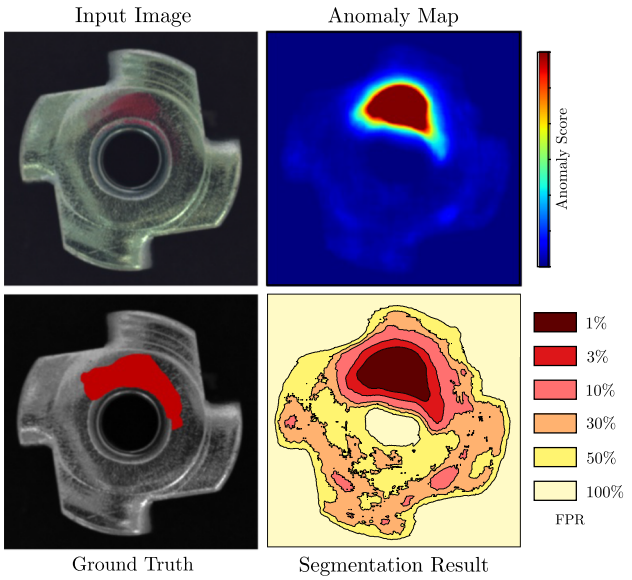

Fig. 4 Example anomaly map for an anomalous input image of class *metal nut*. Binary segmentation results for multiple thresholds are shown as a contour plot. The thresholds are selected such that a certain false positive rate is achieved on the input image. Due to the large class-imbalance between anomalous and anomaly-free pixels, only results at relatively low FPR yield a satisfactory segmentation of the color defect

Five categories cover different types of regular (*carpet*, *grid*) or random (*leather*, *tile*, *wood*) textures, while the remaining ten categories represent various types of objects. Some of these objects are rigid with a fixed appearance (*bottle*, *metal nut*), while others are deformable (*cable*) or include natural variations (*hazelnut*). A subset of objects was acquired in a roughly aligned pose (e.g., *toothbrush*, *capsule*, and *pill*) while others were placed in front of the camera with a random rotation (e.g., *metal nut*, *screw*, and *hazelnut*). The test images of anomalous samples contain a variety of defects, such as defects on the objects' surface (e.g., *scratches*, *dents*), structural anomalies like distorted object parts, or defects that manifest themselves by the absence of certain object parts. In total, 73 different defect types are present, on average five per category. We give a detailed overview of the defects for each category in Table 8. The defects were manually generated with the aim to produce realistic anomalies as they would occur in real-world industrial inspection scenarios. They greatly vary in size, as shown in a box-and-whisker plot (Tukey 1977) in Fig. 3.

All images were acquired using a 2048 × 2048 pixel high-resolution industrial RGB sensor in combination with two bilateral telecentric lenses (Steger et al. 2018, Chapter 2.2.4.2) with magnification factors of 1:5 and 1:1, respectively. Afterwards, the images were cropped to a suitable output size. All image resolutions are in the range between 700 × 700 and 1024 × 1024 pixels. Each dataset image shows a unique physical sample. We did not augment images by taking multiple pictures of the same object in different poses. Since gray-scale images are also common in industrial inspection, three object categories (*grid*, *screw*, and *zipper*) are made available solely as single-channel images. The images were acquired under highly controlled illumination conditions. For some object classes, however, the illumination was altered intentionally to increase variability. We provide pixel-precise ground truth labels for each defective image region. In total, the dataset contains 1888 anomalous regions. All regions were carefully annotated and reviewed by the authors. During the acquisition of the dataset, we generated defects that are confined to local regions, which facilitated a precise labeling of each anomaly. Additionally, pixels on the border of anomalies or lying in ambiguous regions were preferably labelled as anomalous. For locally deformed objects, annotations were created on the deformed area as well as in the region where the deformed object part is expected to be located. Some defects manifest themselves as missing parts. In these cases, we annotated the expected location of the part as anomalous. Some examples of labels for selected anomalous images are displayed in Figs. 1 and 7.

4 Performance Metrics

Assessing the performance of anomaly segmentation algorithms is challenging. In the following, we give an overview of commonly used metrics and discuss the advantages and disadvantages of applying them to the evaluation of anomaly segmentation methods on the proposed dataset. A quantitative comparison of the described metrics can be found in Sect. 6.

In the present work, we study anomaly segmentation algorithms that are capable of returning a real-valued anomaly score for each pixel in a test image. Larger values shall indicate a higher likelihood of a pixel to be anomalous. Let us consider a test set $T := \{I_1, \ldots, I_n\}$ of n images. We denote the anomaly scores for a test image I_i at pixel p as $A_i(p) \in \mathbb{R}$. For each test image, there exists a pixel-precise ground truth $G_i(p) \in \{0, 1\}$ that indicates whether an anomaly is present, i.e., $G_i(p) = 1$, or not, i.e., $G_i(p) = 0$. In order to compare the anomaly scores with the ground truth data, it is necessary to pick a threshold $t \in \mathbb{R}$ to make a binary decision. A pixel is predicted to be anomalous if and only if $A_i(p) > t$. Figure 4 shows an exemplary anomaly map generated by one of the evaluated methods for an anomalous input image of class *metal nut*. It further depicts the corresponding ground truth of the color defect as well as the binary segmentation results for decreasing thresholds as a contour plot.

4.1 Pixel-Level Metrics

Evaluating the performance of anomaly segmentation algorithms on a per-pixel level treats the classification outcome of each pixel as equally important. A pixel can be classified as either a true positive (TP), false positive (FP), true negative (TN), or false negative (FN). For each of the four cases the total number of pixels on the test dataset T is computed as:

$$\mathrm{TP} = \sum_{i=1}^{n} \big| \{p \mid G_i(p) = 1\} \cap \{p \mid A_i(p) > t\} \big|, \tag{1}$$

$$\mathrm{FP} = \sum_{i=1}^{n} \big| \{p \mid G_i(p) = 0\} \cap \{p \mid A_i(p) > t\} \big|, \tag{2}$$

$$\mathrm{TN} = \sum_{i=1}^{n} \big| \{p \mid G_i(p) = 0\} \cap \{p \mid A_i(p) \leq t\} \big|, \tag{3}$$

$$\mathrm{FN} = \sum_{i=1}^{n} \big| \{p \mid G_i(p) = 1\} \cap \{p \mid A_i(p) \leq t\} \big|. \tag{4}$$

where $|S|$ denotes the cardinality of a set S. Based on these absolute measures, which depend on the total number of pixels in the dataset, relative scores such as the per-pixel true positive rate (TPR), false positive rate (FPR), and precision (PRC) can be derived:

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{5}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}, \tag{6}$$

$$\mathrm{PRC} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}. \tag{7}$$

Apart from these three widely used metrics, another common measure to benchmark segmentation algorithms is the intersection over union (IoU), computed on two sets of pixels. In the context of anomaly segmentation, one considers the set of all anomalous predictions, i.e., $P = \bigcup_{i=1}^{n} \{p \mid A_i(p) > t\}$, and the set of all ground truth pixels that are labeled as anomalous, i.e., $G = \bigcup_{i=1}^{n} \{p \mid G_i(p) = 1\}$. Analogously to the relative measures above, the IoU for the class 'anomalous' can also be expressed in terms of absolute pixel classification measures:

$$\mathrm{IoU} = \frac{|P \cap G|}{|P \cup G|} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}. \tag{8}$$

All these measures have the advantage that they are easy and efficient to compute. However, treating each pixel as entirely independent introduces a bias towards large anomalous regions. Detecting a single defect with a large area can make up for the failure to detect numerous smaller defects. Since the size of defects varies greatly for each of the categories in the proposed dataset (cf. Fig. 3), one should further consider metrics that are computed for each connected component of the ground truth.

4.2 Region-Level Metrics

Instead of treating every pixel independently, region-level metrics average the performance over each connected component of the ground truth. This is especially useful if the detection of smaller anomalies is considered equally important as the detection of larger ones. This is often the case in practical applications. In this work, we evaluate the per-region overlap (PRO) that has previously been used to benchmark anomaly segmentation algorithms (Napoletano et al. 2018; Bergmann et al. 2020).

First, for each test image the ground truth is decomposed into its connected components. Let $C_{i,k}$ denote the set of pixels marked as anomalous for a connected component k in the ground truth image i and P_i denote the set of pixels predicted as anomalous for a threshold t. The per-region overlap can then be computed as

$$\mathrm{PRO} = \frac{1}{N} \sum_{i} \sum_{k} \frac{|P_i \cap C_{i,k}|}{|C_{i,k}|}, \tag{9}$$

where N is the number of total ground truth components in the evaluated dataset. The PRO metric is closely related to the TPR. The crucial difference is that the PRO metric averages the TPR over each ground truth region instead of averaging over all image pixels. Note that it is not straightforward to adapt other per-pixel measures such as the PRC or the IoU to the per-region case. This is caused by the fact that they make use of the FPR, and false positives cannot be readily attributed to any specific ground truth region.

4.3 Threshold-Independent Metrics

All of the metrics listed above depend on the previous selection of a suitable threshold t, which is a challenging problem in practice (cf. Sect. 5). If the threshold determination fails, the performance metrics might give a skewed picture of the real performance of a method. Therefore, one often evaluates the above metrics at multiple distinct thresholds. Furthermore, it is desirable to compare two metrics simultaneously since, for example, a high TPR is only useful if the corresponding FPR is low. A way to achieve this is to plot two metrics against each other and compute the area under the resulting curve. A well-known example is the receiver operator characteristic (ROC), which plots the FPR versus the TPR. Another frequently used measure is the precision–recall curve (PR), which plots the true positive rate (recall) versus the precision. In this work, we additionally investigate the

Table 2 Area under the precision–recall curve for each dataset category

Category	f-AnoGAN	Feature dictionary	Student teacher	ℓ_2-autoencoder	SSIM-autoencoder	Texture inspection	Variation model
Carpet	0.025	0.679	**0.711**	0.042	0.035	0.568	0.017
Grid	0.050	0.213	**0.512**	0.252	0.081	0.179	0.096
Leather	0.156	0.276	0.490	0.089	0.037	**0.603**	0.072
Tile	0.093	0.692	**0.789**	0.093	0.077	0.187	0.218
Wood	0.159	0.421	**0.617**	0.196	0.086	0.529	0.213
Bottle	0.160	**0.814**	0.775	0.308	0.309	0.285	0.536
Cable	0.098	**0.617**	0.592	0.108	0.052	0.102	0.084
Capsule	0.033	0.157	**0.377**	0.276	0.128	0.071	0.226
Hazelnut	0.526	0.404	0.585	0.590	0.312	**0.689**	0.485
Metal nut	0.273	0.760	**0.940**	0.416	0.359	0.153	0.384
Pill	0.121	0.724	**0.734**	0.255	0.233	0.207	0.274
Screw	0.062	0.017	**0.358**	0.147	0.050	0.052	0.138
Toothbrush	0.133	0.477	**0.567**	0.367	0.183	0.140	0.416
Transistor	0.130	0.364	0.346	**0.381**	0.191	0.108	0.309
Zipper	0.027	0.369	0.588	0.095	0.088	**0.611**	0.038
Mean	0.136	0.466	**0.599**	0.241	0.148	0.299	0.234

The best-performing method for each dataset category is highlighted in boldface. Overall, methods that leverage pretrained feature extractors for anomaly segmentation outperform all other evaluated approaches

PRO curve, which plots the FPR versus the PRO, as well as the IoU curve, which shows the FPR versus the IoU.

It is important to note that the test split of our anomaly detection dataset is highly imbalanced in the sense that the number of anomalous pixels is significantly smaller than the number of anomaly-free ones. Only 2.7% of all pixels in the test set are labeled as anomalous. Therefore, thresholds that yield a large FPR result in segmentation results that are no longer meaningful. This is especially the case for industrial applications. There, large false positive rates would lead to a large amount of defect-free parts being wrongly rejected. An example is shown in Fig. 4, where segmentation results are given for multiple thresholds as a contour plot. The thresholds were selected such that they result in different false positive rates on the input image, ranging from 1%, for which the defect is well detected, to 100%, where the entire image is segmented as anomalous. For FPRs as low as 30% the segmentation result is already degenerated. Therefore, we additionally include metrics in our evaluations that compute the area under the curves only up to a certain false positive rate. To ensure that the maximum attainable values of this performance measure is equal to 1, we normalize the resulting area. Since the PR curve has been specifically designed to handle large class imbalances and does not use the FPR in its computation, we always evaluate its entire area.

5 Threshold Selection

Evaluating anomaly segmentation algorithms using threshold-independent metrics such as measuring the area under a curve entirely circumvents the need for picking a suitable threshold. However, when employing an algorithm in practice, one must ultimately decide on a threshold value to determine whether a part is classified as defective or not. This is a challenging problem due to the lack of anomalous samples during training time. Even if a small number of anomalous samples was available for threshold estimation, we still consider it preferable to estimate a threshold solely on anomaly-free data. This is because there is no guarantee that the provided samples cover the entire range of possible anomalies and the estimated threshold might perform poorly for other, unknown, types. Instead, we want to find a threshold that separates the distribution of anomaly-free data from the rest of the entire data manifold such that even subtle deviations can be detected.

In this work, we consider three threshold estimation techniques for anomaly segmentation where the thresholds are estimated solely on a set of anomaly-free validation images prior to testing. In our experiments, we evaluated how well each technique transfers from the validation to the test set and which performance is ultimately achieved when selecting these particular thresholds.

Maximum Threshold In theory, a method should classify all pixels of the validation images as anomaly-free. To achieve this, one can simply select the threshold to equal the maxi-

mum value of all occurring anomaly scores on the validation set. In practice, this is often a highly conservative estimate since already a single outlier pixel with a large anomaly score can lead to thresholds that do not perform well on the test set.

p-Quantile Threshold To make the estimation more robust against outliers, one can compute a threshold taking the entire distribution of validation anomaly scores into account and allowing for a certain amount of outlier pixels. Here, we investigate the p-quantile, which selects a threshold such that a percentage p of validation pixels is classified as anomaly-free.

k-Sigma Threshold A third approach is to first compute the mean μ and standard deviation σ over all anomaly scores of the validation set, and then define a threshold to be $t = \mu + k\sigma$. This additionally takes the spread of the distribution of anomaly scores into account. If this distribution can be assumed to perfectly follow a Gaussian distribution, k can also be chosen to achieve a certain false positive rate on the validation set. However, since in practice this might not be the case, the false positive rate on the validation set might differ significantly.

Max-Area Threshold All estimators presented so far compute thresholds simply on the one-dimensional distribution of validation anomaly scores and do not take the spatial location of image pixels into account. In particular, they are insensitive to the size of false positive regions, as many small regions are treated equally to a single larger one. In applications where only anomalies of a certain minimum size are expected, one can leverage this information to filter such small false positive regions and determine a threshold by permitting connected components on the validation images that do not exceed a predefined maximum permissible area. This ensures that an anomaly detector that classifies connected components as anomalous based on their area would not detect a single defect on the validation images.

6 Benchmark

We conduct a thorough evaluation of multiple state-of-the-art methods for unsupervised anomaly segmentation on our dataset. It is intended to serve as a baseline for future methods. We then discuss the strengths and weaknesses of each method on the various objects and textures of the dataset. We show that, while each method can detect anomalies of certain types, none of the evaluated methods manages to excel for the entire dataset. In particular, we find that methods that leverage features of networks pretrained on the ImageNet dataset outperform all other evaluated approaches. Deep learning-based generative models that are trained from scratch, such as convolutional autoencoders or generative adversarial networks, show large room for improvement.

We assess the effect of different performance metrics on the evaluation result and compare different threshold estimation techniques. Furthermore, we provide information on inference time and memory consumption for each evaluated method.

6.1 Training and Evaluation Setup

The following paragraphs list the training and evaluation protocols for each method. For each dataset category, we randomly split 10% of the anomaly-free training images into a validation set. The same validation set was used for all evaluated methods.

Fast AnoGAN For the evaluation of Fast AnoGAN (f-AnoGAN), we use the publicly available implementation by the original authors on Github.[3] The GAN's latent space dimension is fixed to 128 and generated images are of size 64 $\times$ 64 pixels, which results in a relatively stable training for all categories of the dataset. GAN training is conducted for 100 epochs using the Adam optimizer with an initial learning rate of 10^{-4} and a batch size of 64. The encoder network for fast inference is trained for 50 000 iterations using the RMSProp optimizer with an initial learning rate of 5×10^{-5} and batch size of 64. Since the implementation of Fast AnoGAN only operates on single-channel images, all input images are converted to grayscale beforehand.

Anomaly maps are obtained by a per-pixel ℓ^2 - comparison of the input image with the generated output. For all evaluated dataset categories, training, validation and testing images are zoomed to size 256 $\times$ 256 pixels. 50 000 training patches of size 64 $\times$ 64 pixels are randomly cropped from the training images. During testing, a patchwise evaluation is performed with a horizontal and vertical stride of 64 pixels.

ℓ^2- and SSIM-Autoencoder:

For the evaluation of the ℓ^2- and SSIM-autoencoder, we build on the same CAE architecture that was described by Bergmann et al. (2019b). They reconstruct patches of size 128 $\times$ 128, employing either a per-pixel ℓ^2 loss or a loss based on the structural similarity index (SSIM). We extend the architecture by an additional convolution layer to process images at resolution 256 $\times$ 256. We find an SSIM window size of 11 $\times$ 11 pixels to work well in our experiments. The latent space dimension is chosen to be 128. Larger latent space dimensions do not yield significant improvements in reconstruction quality while lower dimensions lead to degenerate reconstructions. Training is run for 100 epochs using

[3] https://github.com/tSchlegl/f-AnoGAN.

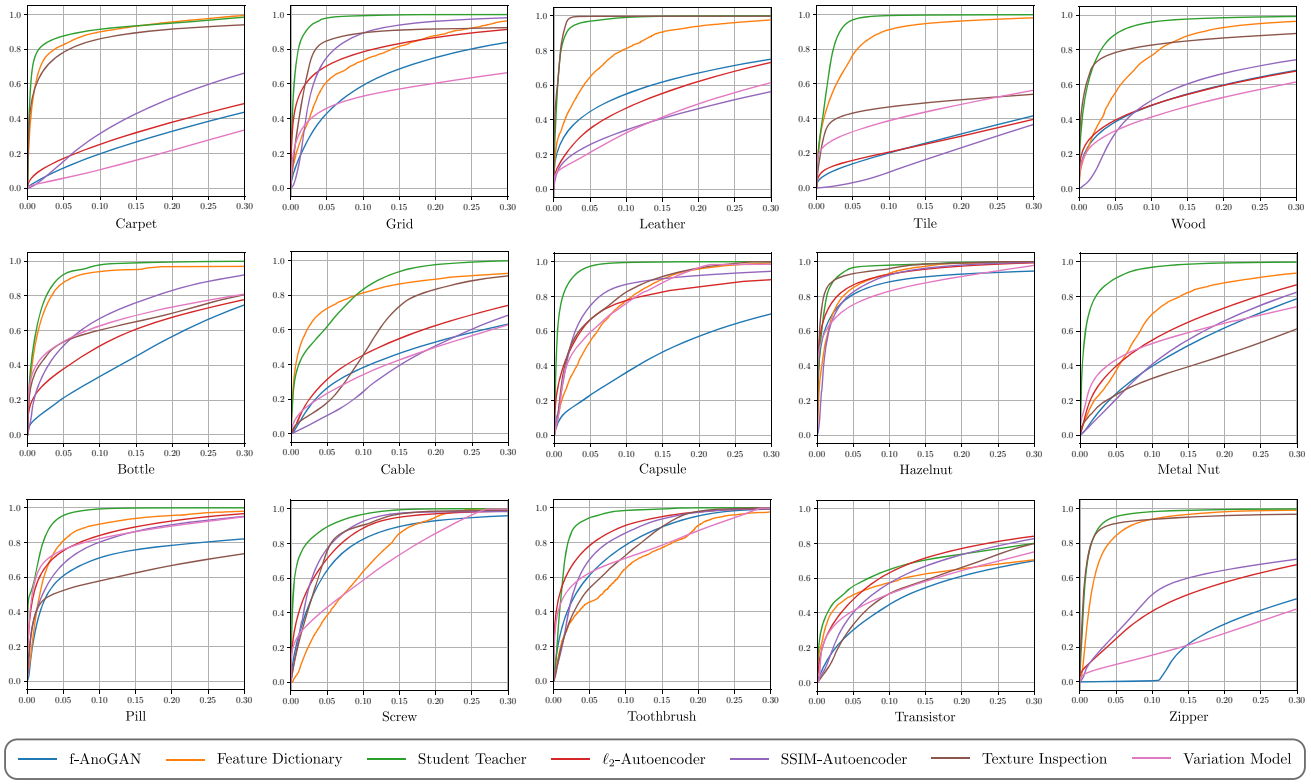

Fig. 5 PRO curves for each dataset category and all evaluated methods. The per-region overlap (y-axis) is plotted against false positive rates up to 30% (x-axis)

Table 3 Comparison of threshold independent performance metrics

Metric	f-AnoGAN	Feature dictionary	Student teacher	ℓ^2- Autoencoder	SSIM-autoencoder	Texture inspection	Variation model
AU-PR	0.136 (7)	0.466 (2)	**0.599 (1)**	0.241 (4)	0.148 (6)	0.299 (3)	0.234 (5)
AU-ROC	0.472 (7)	0.836 (2)	**0.922 (1)**	0.590 (4)	0.584 (5)	0.656 (3)	0.526 (6)
AU-PRO	0.528 (7)	0.803 (2)	**0.924 (1)**	0.632 (4)	0.617 (5)	0.741 (3)	0.556 (6)
AU-IoU	0.073 (7)	0.168 (2)	**0.190 (1)**	0.099 (4)	0.091 (6)	0.100 (3)	0.095 (5)
$\text{AU-PRO}_{0.01}$	0.113 (6)	0.201 (4)	**0.432 (1)**	0.218 (3)	0.075 (7)	0.263 (2)	0.197 (5)
$\text{AU-PRO}_{0.05}$	0.249 (7)	0.459 (3)	**0.734 (1)**	0.372 (4)	0.279 (6)	0.488 (2)	0.328 (5)
$\text{AU-PRO}_{1.00}$	0.784 (7)	0.931 (2)	**0.974 (1)**	0.838 (5)	0.840 (4)	0.890 (3)	0.796 (6)

For each metric and evaluated method, the normalized area under the curve is computed and averaged across all dataset categories. The best-performing method for each dataset category is highlighted in boldface. The ranking of each method with respect to the evaluated metric is given in brackets. For the ROC, PRO and IoU curves, the area is computed up to an FPR of 30%. The AU-PRO metric is additionally reported for varying integration limits

the Adam optimizer with an initial learning rate of 2×10^{-4} and a batch size of 128.

For each dataset category, 10 000 training samples are augmented from the train split of the original dataset. For textures, randomly sampled patches are cropped evenly across the training images. For objects, we apply a random translation and rotation to the entire input image and zoom the result to match the autoencoder's input resolution. Additional mirroring is applied where the object permits it.

For the dataset objects, anomaly maps are generated by passing an image through the autoencoder and comparing the reconstruction with its respective input using either per-pixel ℓ^2 comparisons or SSIM. For textures, we reconstruct patches at a stride of 64 pixels and average the resulting anomaly maps. Since SSIM does not operate on color images, for the training and evaluation of the SSIM-autoencoder all images are converted to grayscale.

Feature Dictionary We use our own implementation of the CNN feature dictionary proposed by Napoletano et al. (2018), which extracts features from the 512-dimensional average pooling layer of a ResNet-18 pretrained on ImageNet. Principal Component Analysis (PCA) is performed on the extracted features to explain 95% of the variance. K-means is run with 50 cluster centers and the nearest descriptor to each center is stored as a dictionary vector. We extract 100 000 patches of size 128 × 128 for both the textures and objects. All images are evaluated at their original resolution. A stride of 8 pixels is chosen to create a spatially resolved anomaly map. For grayscale images, the channels are triplicated for feature extraction since the used ResNet-18 operates on three-channel input images.

Student–Teacher Anomaly Detection For the evaluation of Student–Teacher anomaly detection, we use three teacher networks pretrained on ImageNet that extract dense feature maps at receptive field sizes of 17, 33, and 65 for an input image size of 256 × 256 pixels. For each teacher network, an ensemble of 3 student networks is trained to regress the teacher's output. We use the same network architectures as described by Bergmann et al. (2020). For the pretraining of teacher networks, we follow the proposed training protocol by the original authors, using only the feature matching and correlation loss for knowledge distillation. Training is performed for 100 epochs using the Adam optimizer at an initial learning rate of 10^{-4} and a batch size of 1.

GMM-Based Texture Inspection Model For the Texture Inspection Model (Böttger and Ulrich 2016), an optimized implementation is available in the HALCON machine vision library.[4] Images are converted to grayscale, zoomed to an input size of 400 × 400 pixels, and a four-layer image pyramid is constructed for training and evaluation. On each pyramid level, a separate GMM with dense covariance matrix is trained. The patch size of examined texture regions on each pyramid level is set to 7 × 7 pixels. We use a maximum of 50 randomly selected images from the original training set for training the Texture Inspection Model. Anomaly maps for each pyramid level are obtained by evaluating the negative log-likelihood for each image pixel using the corresponding trained GMM. We normalize the anomaly scores of each level such that the mean score is equal to 0 and their standard deviation equal to 1 on the validation set. The different levels are then combined to a single anomaly map by averaging the four normalized anomaly scores per pixel position.

Variation Model In order to create the Variation Model (Steger et al. 2018, Chapter 3.4.1.4), we use all available training images of each dataset category in their original size and calculate the mean and standard deviation at each pixel location. This works best if the images show aligned objects. Since this is not always the case, we implemented a specific alignment procedure for our experiments on the following six dataset categories. *Bottle* and *metal nut* are aligned using shape-based matching (Steger2001 2001; Steger 2002), *grid* and *transistor* using template matching with normalized cross-correlation as the similarity measure (Steger et al. 2018, Chapter 3.11.1.2). *Capsule* and *screw* are segmented via simple thresholding and then aligned by using a rigid transformation which is determined by geometric features of the segmented region.

The anomaly map for a test image is obtained as follows. We define the value of each pixel in the anomaly map by calculating the distance from the gray value of the corresponding test pixel to the trained mean value and divide this distance by a multiple of the trained standard deviation. For multichannel images, this process is done separately for each channel and we obtain an overall anomaly map as the pixelwise maximum of all the channels' individual maps. Note that when a spatial transformation is applied to input images during inference, some input pixels might not overlap with the mean and deviation images. For such pixels, no meaningful anomaly score can be computed. In our evaluation, we set the anomaly score for such pixels to the minimum attainable value of 0. As for the GMM-based Texture Inspection, we use the optimized implementation of the HALCON machine vision library.

6.2 Performance Evaluation

We begin by comparing the performance of all methods for different threshold independent evaluation metrics, followed by an analysis of each method individually. The computation of curve areas that involve the false positive rate is performed up to an FPR of 0.3 if not mentioned otherwise.

Table 2 shows the area under the precision–recall curve for each method and dataset category. The Student–Teacher anomaly detection method performs best for most of the evaluated objects and textures. Regarding each method's mean performance on the dataset, the two top-performing approaches, i.e., Student–Teacher and Feature Dictionary, both leverage pretrained feature extractors. The generative deep learning-based methods that are trained from scratch perform significantly worse, often only performing on par or inferior to the more traditional approaches, i.e., the Variation Model and the GMM-based Texture Inspection. For each object and method, the corresponding PRO curves are given in Fig. 5. Benchmark results for all other evaluated metrics on all dataset categories can be found in Appendix A.

Table 3 assesses the influence of different performance metrics on the evaluation result. The mean area under the ROC, PRO, PR, and IoU curves are given for each evalu-

[4] https://www.mvtec.com/products/halcon.

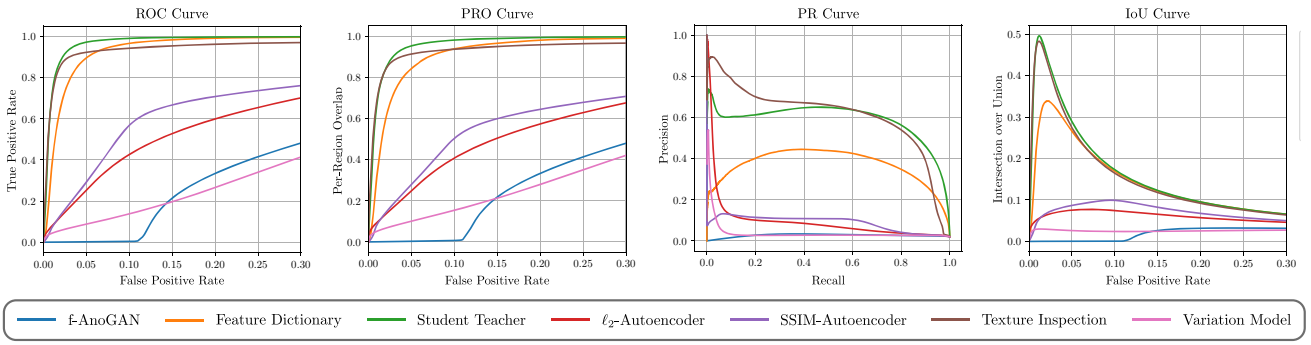

Fig. 6 Comparison of performance curves for the dataset category *zipper*

ated method. Areas are averaged over all dataset categories. Additionally, the area under the PRO curve is computed up to three different integration limits: 0.01, 0.05, and 1.0. For each method, its ranking with respect to the current metric and all other methods is given in brackets. When integrating up to a false positive rate of 30%, (first three rows) the rankings produced by the investigated metrics are fairly consistent. Student–Teacher anomaly detection performs best in all cases, followed by the CNN Feature Dictionary. f-AnoGAN performs worst for all evaluated metrics. When varying the integration threshold of the false positive rate for the AU-PRO metric (last three rows), the ranking of some methods changes significantly. For example, when evaluating the full area under the PRO curve, the CNN Feature Dictionary ranks second, while it ranks only fourth place if the area is computed only up to an FPR of 1%. This highlights that the choice of the integration threshold is important for metrics that involve the false positive rate and one must select it carefully depending on the requirements of the application. For tasks where low false positive rates are crucial, one might prefer sufficiently small integration thresholds over larger ones.

Table 3 further shows that when decreasing the integration limit of the FPR, the area under the PRO curve drops for all methods within factors of 2 (Student–Teacher) and 11 (SSIM-Autoencoder). This shows that many methods only manage to detect anomalies when at the same time a significant amount of false positive pixels are allowed in the segmentation result. This might limit the applicability of these methods in practice, as illustrated in Fig. 4. Figure 6 shows example curves for all evaluated metrics for the dataset category *zipper*. For this object, defect sizes do not vary as much as for other dataset categories (Fig. 3), and hence the ROC and PRO curves are similar. The PR curve shows that the precision of all methods except Student–Teacher and the Texture Inspection Model is smaller than 0.5 for most recall values. This indicates that these methods predict more false positive pixels than true positives for any threshold. Compared to the precision, the IoU additionally takes the false negative predictions into account. Therefore, the IoU is bounded by the maximum attained precision value and methods with low overall precision also yield low IoU values for any threshold. For large false positive rates, the IoU converges towards the ratio of the number of ground truth anomalous pixels divided by the total number of pixels in the evaluated dataset.

Figure 7 shows an example for each method where anomaly detection worked well, i.e., the thresholded anomaly map substantially overlaps with the ground-truth (left column) and where each method produced an unsatisfactory result (right column). Anomaly scores were thresholded such that an average FPR of 0.01 across the entire test set is achieved. Based on the selected images, we now discuss individual properties of each evaluated method when applied to our dataset.

f-AnoGAN The f-AnoGAN method computes anomaly scores based on per-pixel comparisons between its input and reconstruction. Due to the increased contrast between the screw and the background, it manages to segment the tip of the screw. Because of imperfect reconstructions, however, the method also yields numerous false positives around the objects' edges and around regions where strong reflections are present. It entirely fails to detect the structural anomaly on the carpet since a removal of the defect by reconstruction does not result in an image substantially different from the input.

Feature Dictionary The CNN Feature Dictionary was originally designed to model the distribution of repetitive texture patches. However, it also yields promising results for anomaly segmentation on objects when anomalies manifest themselves in features that deviate strongly from the local descriptors of the training data manifold. For example, the small crack on the capsule is well detected. However, since the method randomly subsamples training patches, it yields increased anomaly scores in regions that are underrepresented in the training set, e.g., on the imprint on the left half

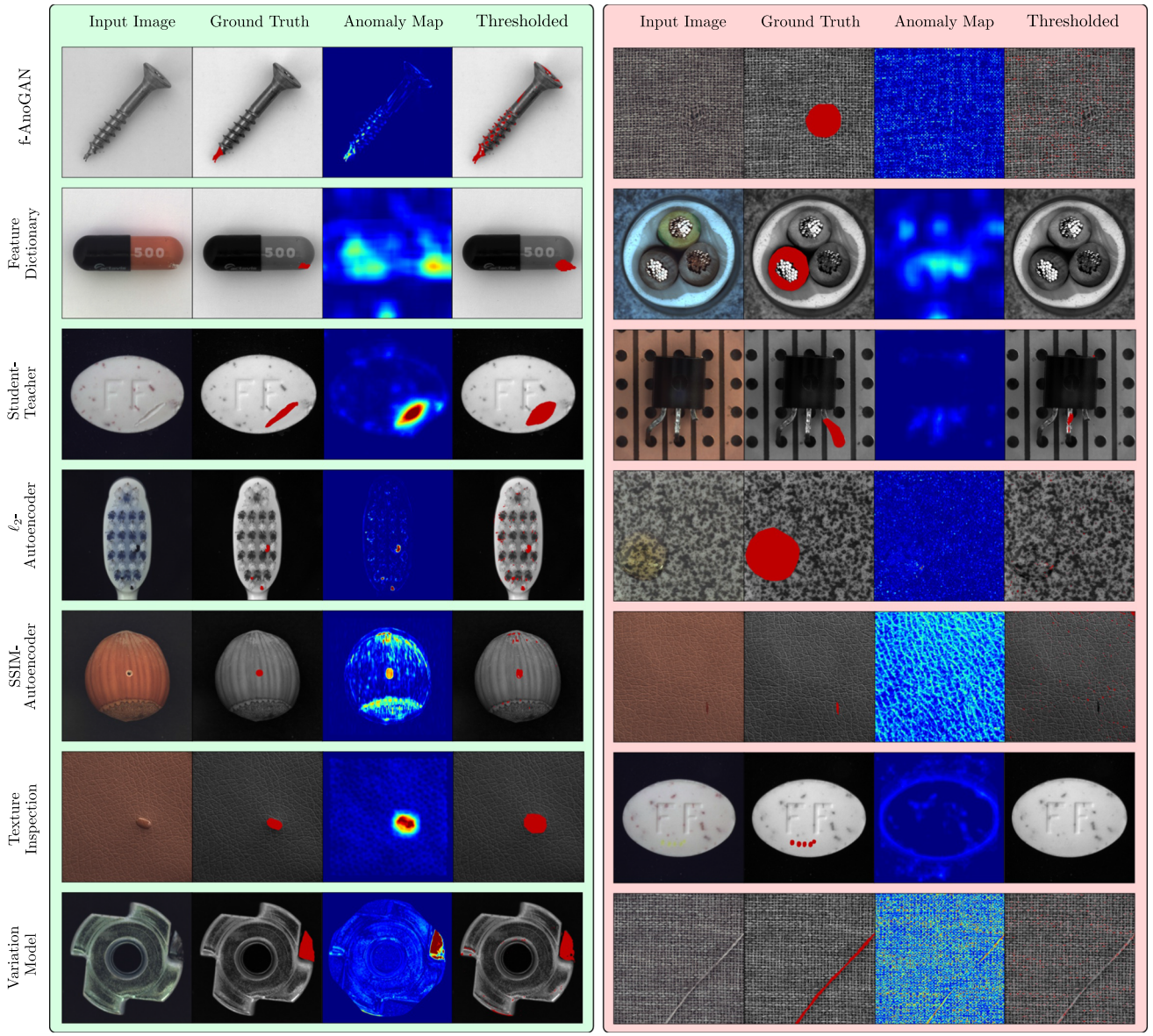

Fig. 7 Qualitative results for each evaluated method. The left column shows examples where each method worked well. A failure case is shown in the right column. Thresholds were selected such that a false positive rate of 0.01 is achieved on the test set of an evaluated category

of the capsule. Additionally, due to the limited capacity of K-Means, the training feature distribution is often insufficiently well approximated. The method does not capture the global context of an object. Hence, it fails to detect the anomaly on the cable cross section, where the inner insulation on the bottom left shows the wrong color, as it is brown instead of blue.

Student–Teacher Anomaly Detection This method exhibited the best overall performance in our benchmark. Similarly to the CNN Feature Dictionary, the Student–Teacher approach models the distribution of local patch descriptors. It outputs dense anomaly maps with an anomaly score for each input pixel and hence does not require a strided evaluation that might lead to rather coarse segmentations. Since this approach does not rely on data subsampling during training and makes use of all available training features, its anomaly scores show only small variations in anomaly-free regions. However, one can observe slightly increased anomaly scores on the transistor's legs, which exhibit strong reflections and make feature prediction challenging. Like the CNN Feature Dictionary, it does not incorporate global context and therefore fails to detect the missing leg of the transistor.

ℓ^2- and SSIM-Autoencoder Both autoencoders rely on accurate reconstruction of their inputs for precise anomaly

detection. However, they often fail to reconstruct small details and produce blurry images. Therefore, they tend to yield increased anomaly scores in regions that are challenging to reconstruct accurately, as can be observed on the object boundaries of the hazelnut and the bristles of the toothbrush. Like for f-AnoGAN, the ℓ^2-autoencoders per-pixel comparisons result in unsatisfactory anomaly segmentation performance when the gray-value difference is small between the input and reconstruction, as is the case for the transparent color defect on the tile. Since the SSIM-autoencoder only operates on grayscale images, it often fails to detect color defects entirely, such as the red color stroke on the leather texture.

GMM-Based Texture Inspection Model HALCON's Texture Inspection models the distribution of gray-values within local image patches using a GMM. It performs well on uniform texture patterns, for example, those that are present in the dataset category leather. Since it only operates on grayscale images, it often fails to detect color defects such as the one on the pill. Because the boundaries of objects are underrepresented in the training data, it often yields increased anomaly scores in these areas.

Variation Model For the evaluation of the Variation Model, prior object alignment is performed where possible. It performs well for rigid objects such as the metal nut, which allows for a precise alignment. Due to the applied transformation, not every single input image pixel overlaps with the mean and deviation image. For these background pixels, no meaningful anomaly score can be computed. For dataset categories where an alignment is not possible, e.g., *carpet*, this method fails entirely. Due to the high variance of the gray values in the training images, the model assigns high likelihoods to almost every gray value.

6.3 Threshold Estimation Techniques

In Sect. 5, we discussed various techniques to estimate thresholds purely on a validation set of anomaly-free images. In order to assess their performance in practice, we computed thresholds on three different categories of the dataset: *bottle*, *pill*, and *wood*. The Maximum threshold simply selects the maximum anomaly score of all validation pixels. For the p-Quantile threshold, we used $p = 0.99$, which means that one percent of all validation pixels will be marked as anomalous by each method. We selected a k-Sigma threshold such that under the assumption of normally distributed anomaly scores, also a quantile of 0.99 is reached. We additionally investigated a Max-Area threshold that allows connected components of anomalous pixels with an area smaller than 0.1% of the area of the entire input image.

Figure 8 marks the FPR and PRO values achieved when applying the different thresholds. For each dataset category, the three best performing methods in terms of AU-PRO are displayed. Since the Maximum threshold does not allow a single false positive pixel on the entire validation set, it is the most conservative threshold estimator among the evaluated ones, yielding the lowest false positive rates on the test set. However, in some cases, it entirely fails to produce any true positives as well, due to outliers on the validation set.

All other threshold estimation techniques allow a certain amount of false positives on the validation set. Hence, they also yield increased false positive rates on the test set. Both The p-Quantile and k-Sigma thresholds attempt to fix the false positive rate at one percent. However, due to the inaccurate segmentations of each method, the application of each threshold results in a significantly higher FPR. Furthermore, the marker locations of the two thresholds are often very different for the same anomaly detection method, which indicates that the assumption of normally distributed anomaly

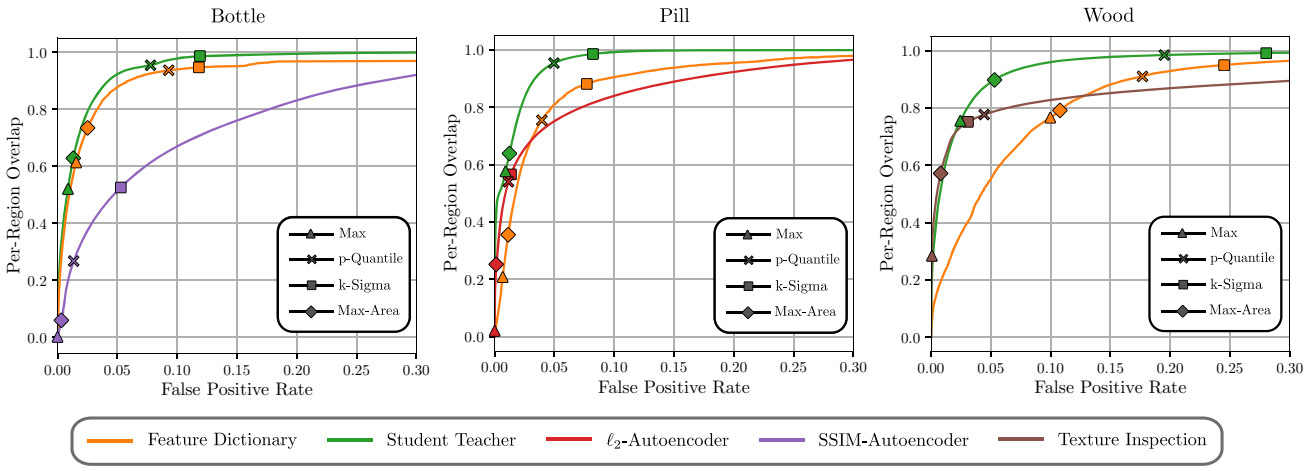

Fig. 8 Performance of different threshold estimates in terms of FPR and PRO on three different dataset categories. For each category, the three top performing methods are displayed. Thresholds are computed on a validation set of anomaly-free images

scores often does not hold in practice. For many of the evaluated methods, the Max-Area threshold is only slightly less conservative than picking the maximum of all anomaly scores. This indicates that already only a slight decrease of the Maximum threshold results in connected components of false positives that one might deem large enough to classify them as anomalies in practice.

Our results show that selecting a suitable threshold for anomaly segmentation purely on anomaly-free validation images is a highly challenging problem in practice. The same estimator might yield very different results depending on the anomaly detection method and dataset under consideration. For applications that require very low false positive rates, one is at risk of picking too conservative thresholds that fail to detect any anomalies. On the other hand, allowing for too many false positives quickly yields segmentation results that are no longer useful in practice as well.

6.4 Time and Memory Consumption

The runtime and required memory of a method during inference are important criteria for its applicability in real-world scenarios. However, since both greatly depend on implementation-specific details, measuring them accurately is challenging. For example, the amount of memory used by deep-learning based methods can often be greatly reduced when freeing intermediate feature maps during a forward pass. The execution time of an algorithm is directly affected by the specific libraries being used and the amount of exploited potential for parallelization. Hence, we do not provide exact numbers for inference time and memory consumption but rather point out qualitative differences between the evaluated methods.

As can be expected, the methods performing multiple forward passes through a network have the highest inference times. A particularly extreme example is the CNN Feature Dictionary, which requires several seconds to process a single image. This is due to the patch-wise evaluation and the fact that only a single anomaly score is produced for each patch. It is possible to reduce the time by using a larger stride for the patches at the cost of coarser anomaly maps. Methods that require only a single model evaluation per image and run entirely on the GPU, such as the autoencoders evaluated on the objects of the dataset, allow for much faster inference times in the range of a few milliseconds. However, when performing strided evaluations on the textures, multiple forward passes become necessary and their runtime increases to several hundreds of milliseconds. The same is true for f-AnoGAN. Since the Student–Teacher anomaly detection employs multiple teacher networks and an ensemble of students for each teacher, the evaluation of a single image requires a forward pass through each of the models. Because the evaluation of each model falls in the range of tens of milliseconds, the total runtime is in the range of several hundreds of milliseconds. For the more traditional methods, i.e., the Variation Model and the Texture Inspection Model, we use optimized implementations of the HALCON machine vision library that entirely run on the CPU and achieve runtimes in the range of tens and hundreds of milliseconds, respectively.

Table 4 Approximate number of model parameters of each evaluated deep learning based method in millions

Method	#Parameters
f-AnoGAN	24.57 M
Feature dictionary	11.46 M
Student teacher	26.07 M
ℓ^2-autoencoder	1.20 M
SSIM-autoencoder	1.20 M

In order to facilitate a relative comparison of the amount of memory required to perform inference in deep learning models, one commonly reports the total number of model parameters as a lower bound. The number of parameters for each model evaluated in this paper is given in Table 4. Since the Variation Model and the Texture Inspection Model are not based on deep learning and work in an entirely different way, simply counting the number of model parameters and comparing them to the deep learning based approaches is not advisable. The Variation Model, for example, stores two model parameters for each image pixel and thus, the total number of parameters is in the same range as one of the evaluated deep learning models. However, the Variation Model does not need to allocate any additional memory and one can still expect the deep learning based approaches to consume a lot more memory during inference due to their intermediate computation of high-dimensional feature maps.

7 Conclusions

We have introduced the MVTec Anomaly Detection dataset, a novel dataset for unsupervised anomaly detection that mimicks real-world industrial inspection scenarios. The dataset provides the possibility to evaluate unsupervised anomaly detection methods on various texture and object classes with different types of anomalies. Because pixel-precise ground truth labels for anomalous regions in the images are provided, it is possible to evaluate anomaly detection methods for both image-level classification as well as pixel-level segmentation.

We have evaluated several state-of-the-art methods as well as two classical methods for anomaly segmentation thoroughly on this dataset. The evaluations are intended to serve as a baseline for the development of future methods.

Our results show that discriminative approaches that leverage descriptors of pretrained networks outperform methods that learn feature representations from scratch solely on the anomaly-free training data. We have provided information on inference time as well as memory consumption for each evaluated method.

Furthermore, we have discussed properties of common evaluation metrics and threshold estimation techniques for anomaly segmentation and have highlighted their advantages and shortcomings. We have shown that determining suitable thresholds solely on anomaly-free data is a challenging problem because the performance of each estimator highly varies for different dataset categories and evaluated methods.

We hope that the proposed dataset will stimulate the development of new unsupervised anomaly detection methods.

Author contributions The first author named is lead and corresponding author. All other authors are listed in alphabetical order. We further list individual contributions to the paper. *Writing—Original Draft:* P.B., K.B., M.F., and D.S. *Writing—Review & Editing:* P.B., K.B., M.F., D.S, and C.S. *Conceptualization:* P.B, K.B, M.F., D.S., and C.S. *Methodology:* P.B, K.B., M.F., and D.S. *Investigation and Data Curation (Aquisition and Annotation of the Dataset):* P.B., M.F., and D.S.

Data Availability Statement The proposed dataset is publicly available for download at https://www.mvtec.com/company/research/datasets.

A Additional Results

For completeness, we provide additional qualitative and quantitative results for each evaluated method and dataset category. Tables 5, 6 and 7 report the area under the ROC, IoU, and PR curves, respectively. Areas were computed up to a false positive rate of 30% and normalized with respect to the maximum attainable value. Figures 9 and 10 qualitatively show anomaly images produced for the different methods on selected dataset categories (Table 8).

Table 5 Normalized area under the ROC curve up to an average false positive rate per-pixel of 30% for each dataset category

Category	f-AnoGAN	Feature dictionary	Student teacher	ℓ_2-autoencoder	SSIM-autoencoder	Texture inspection	Variation model
Carpet	0.251	**0.943**	0.927	0.287	0.365	0.874	0.162
Grid	0.550	0.872	**0.974**	0.741	0.820	0.878	0.488
Leather	0.574	0.819	**0.976**	0.491	0.356	0.975	0.381
Tile	0.180	0.854	**0.946**	0.174	0.156	0.314	0.304
Wood	0.392	0.720	**0.895**	0.417	0.404	0.723	0.408
Bottle	0.422	**0.953**	0.943	0.528	0.624	0.454	0.667
Cable	0.453	0.797	**0.866**	0.510	0.302	0.512	0.423
Capsule	0.362	0.793	**0.952**	0.732	0.799	0.698	0.843
Hazelnut	0.825	0.911	**0.959**	0.879	0.847	0.955	0.802
Metal nut	0.435	0.862	**0.979**	0.572	0.539	0.135	0.462
Pill	0.504	0.911	**0.955**	0.690	0.698	0.440	0.666
Screw	0.814	0.738	**0.961**	0.867	0.885	0.877	0.697
Toothbrush	0.749	0.916	**0.971**	0.837	0.846	0.712	0.775
Transistor	0.372	0.527	0.566	**0.657**	0.562	0.363	0.601
Zipper	0.201	0.921	**0.964**	0.474	0.564	0.928	0.209
Mean	0.472	0.836	**0.922**	0.590	0.584	0.656	0.526

The best-performing method for each dataset category is highlighted in boldface

Table 6 Normalized area under the PRO curve up to an average false positive rate per-pixel of 30% for each dataset category

Category	f-AnoGAN	Feature dictionary	Student teacher	ℓ^2-autoencoder	SSIM-autoencoder	Texture inspection	Variation model
Carpet	0.253	0.895	**0.914**	0.306	0.392	0.855	0.165
Grid	0.626	0.757	**0.973**	0.798	0.847	0.857	0.545
Leather	0.584	0.819	0.971	0.519	0.389	**0.981**	0.394
Tile	0.252	0.873	**0.949**	0.251	0.166	0.472	0.425
Wood	0.517	0.778	**0.929**	0.520	0.530	0.827	0.455
Bottle	0.440	0.906	**0.942**	0.567	0.703	0.636	0.659
Cable	0.428	0.815	**0.840**	0.507	0.368	0.597	0.405
Capsule	0.447	0.791	**0.971**	0.771	0.830	0.834	0.802
Hazelnut	0.872	0.913	**0.961**	0.922	0.897	0.958	0.849
Metal nut	0.482	0.701	**0.943**	0.607	0.501	0.384	0.562
Pill	0.700	0.872	**0.958**	0.847	0.803	0.606	0.834
Screw	0.808	0.725	**0.948**	0.864	0.875	0.864	0.701
Toothbrush	0.809	0.718	**0.946**	0.891	0.841	0.786	0.774
Transistor	0.494	0.590	**0.664**	0.657	0.602	0.542	0.554
Zipper	0.202	0.897	**0.955**	0.457	0.515	0.923	0.221
Mean	0.528	0.803	**0.924**	0.632	0.617	0.741	0.556

The best-performing method for each dataset category is highlighted in boldface

Table 7 Normalized area under the IoU curve up to an average false positive rate per-pixel of 30% for each dataset category

Category	f-AnoGAN	Feature dictionary	Student teacher	ℓ_2-autoencoder	SSIM-autoencoder	Texture inspection	Variation model
Carpet	0.025	**0.139**	**0.139**	0.030	0.034	0.123	0.015
Grid	0.030	0.057	**0.075**	0.050	0.046	0.058	0.032
Leather	0.035	0.051	0.072	0.027	0.019	**0.074**	0.020
Tile	0.057	0.315	**0.361**	0.055	0.044	0.113	0.106
Wood	0.089	0.171	**0.228**	0.096	0.081	0.188	0.096
Bottle	0.115	**0.327**	0.321	0.159	0.187	0.142	0.218
Cable	0.080	0.172	**0.179**	0.087	0.043	0.081	0.069
Capsule	0.024	0.061	**0.086**	0.061	0.061	0.047	0.070
Hazelnut	0.141	0.150	0.168	0.150	0.134	**0.172**	0.133
Metal nut	0.188	0.423	**0.505**	0.262	0.239	0.059	0.217
Pill	0.099	0.220	**0.231**	0.142	0.142	0.100	0.141
Screw	0.020	0.013	**0.032**	0.023	0.021	0.021	0.017
Toothbrush	0.080	0.123	**0.137**	0.104	0.097	0.076	0.099
Transistor	0.095	0.153	0.157	**0.182**	0.143	0.088	0.166
Zipper	0.018	0.148	**0.168**	0.062	0.073	0.162	0.026
Mean	0.073	0.168	**0.190**	0.099	0.091	0.100	0.095

The best-performing method for each dataset category is highlighted in boldface

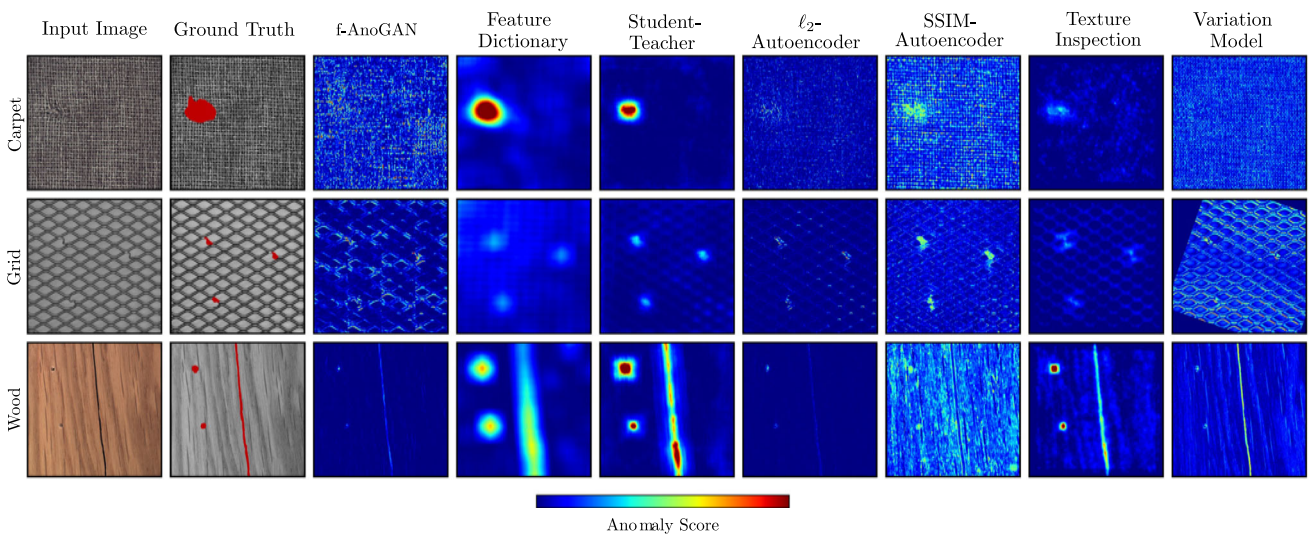

Fig. 9 Additional qualitative results for three selected textures of our dataset

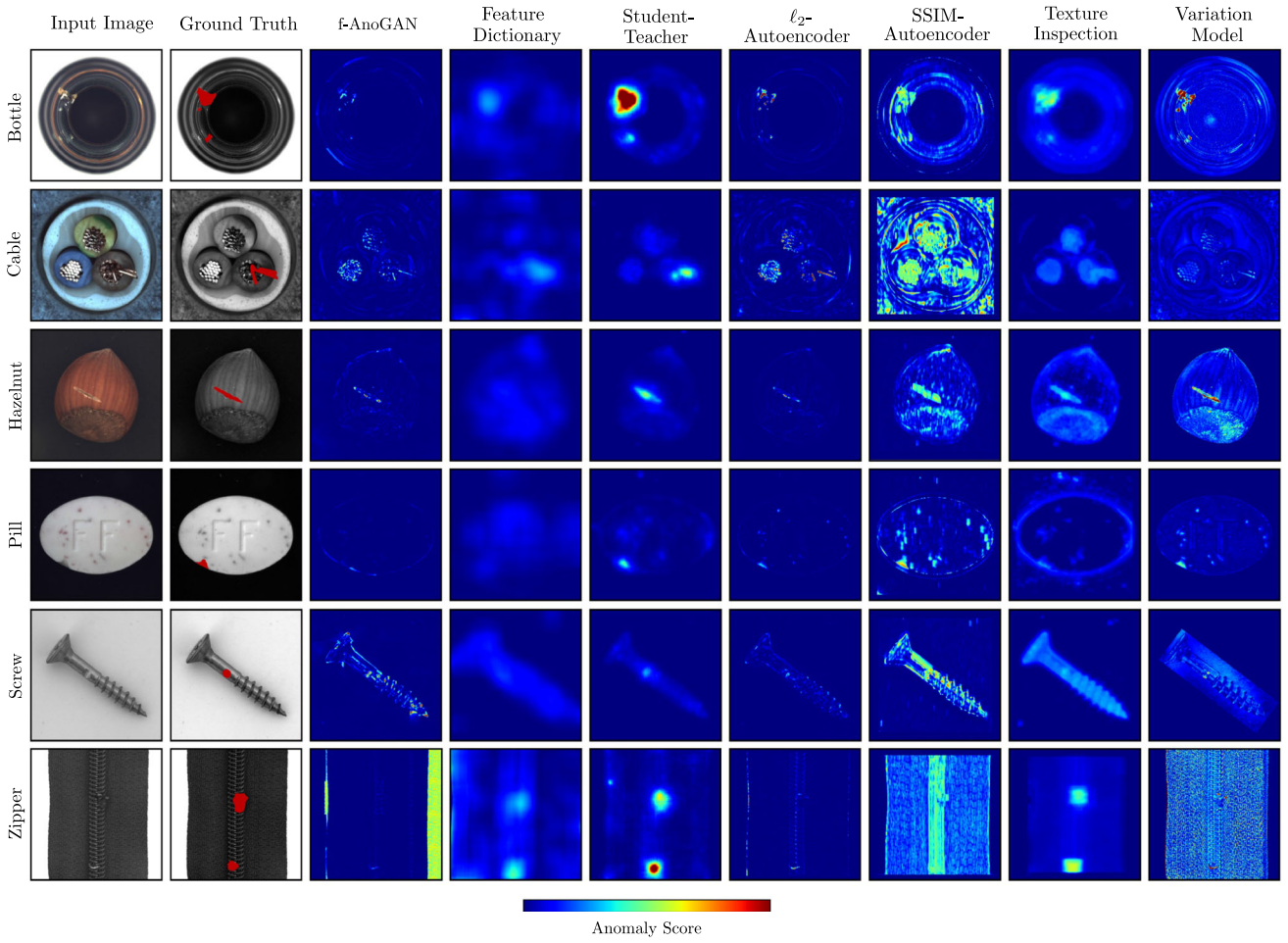

Fig. 10 Additional qualitative results for six selected objects of our dataset

Table 8 Overview over the number of images for each defect type for each category

Category	Defect Type	#Images
Carpet	Color	19
	Cut	17
	Hole	17
	Metal contamination	17
	Thread	19
Grid	bent	12
	Broken	12
	Glue	11
	Metal contamination	11
	Thread	11
Leather	Color	19
	Cut	19
	Fold	17
	Glue	19
	Poke	18
Tile	Crack	17
	Glue strip	18
	Gray stroke	16
	Oil	18
	Rough	15
Wood	Color	8
	Combined	11
	Hole	10
	Liquid	10
	Scratch	21
Bottle	Broken large	20
	Broken small	22
	Contamination	21
Cable	Bent wire	13
	Cable swap	12
	Combined	11
	Cut inner insulation	14
	Cut outer insulation	10
	Missing cable	12
	Missing wire	10
	Poke insulation	10
Capsule	Crack	23
	Faulty imprint	22
	Poke	21
	Scratch	23
	Squeeze	20
Hazelnut	Crack	18
	Cut	17
	Hole	18
	Print	17
Metal nut	Bent	25
	Color	22
	Flip	23
	Scratch	23
Pill	Color	25
	Combined	17
	Contamination	21
	Crack	26
	Faulty imprint	19
	Pill type	9
	Scratch	24
Screw	Manipulated front	24
	Scratch head	24
	Scratch neck	25
	Thread side	23
	Thread top	23
Toothbrush	Defective	30
Transistor	Bent lead	10
	Cut lead	10
	Damaged case	10
	Misplaced	10
Zipper	Broken teeth	19
	Combined	16
	Fabric border	17
	Fabric interior	16
	Rough	17
	Split teeth	18
	Squeezed teeth	16

References

An, J., & Cho, S. (2015). *Variational autoencoder based anomaly detection using reconstruction probability*. SNU Data Mining Center: Tech. rep.

Baur, C., Wiestler, B., Albarqouni, S., & Navab, N. (2019). Deep autoencoding models for unsupervised anomaly segmentation in brain MR images. In A. Crimi, S. Bakas, H. Kuijf, F. Keyvan, M. Reyes, & T. van Walsum (Eds.), *Brainlesion: Glioma, multiple sclerosis, stroke and traumatic brain injuries* (pp. 161–169). Cham: Springer.

Bergmann, P., Fauser, M., Sattlegger, D., Steger, C. (2019a). MVTec AD: A comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 9592–9600).

Bergmann, P., Löwe, S., Fauser, M., Sattlegger, D., Steger, C. (2019b). Improving Unsupervised Defect Segmentation by Applying Structural Similarity to Autoencoders. In: Tremeau A, Farinella G, Braz J (eds) 14th international joint conference on computer vision, imaging and computer graphics theory and applications. Scitepress, Setúbal, vol 5: VISAPP, pp 372–380

Bergmann, P., Fauser, M., Sattlegger, D., Steger, C. (2020). Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 4182–4191).

Blum, H., Sarlin, P. E., Nieto, J., Siegwart, R., Cadena, C. (2019). Fishyscapes: A benchmark for safe semantic segmentation in autonomous driving. In *The IEEE international conference on computer vision (ICCV) workshops*.

Böttger, T., & Ulrich, M. (2016). Real-time texture error detection on textured surfaces with compressed sensing. *Pattern Recognition and Image Analysis*, *26*(1), 88–94.

Bulatov, Y. (2011). notMNIST dataset. Tech. rep. https://yaroslavvb.blogspot.com/2011/09/notmnist-dataset.html.

Burlina, P., Joshi, N., & Wang, I. J. (2019). Where's Wally now? Deep generative and discriminative embeddings for novelty detection. In *IEEE conference on computer vision and pattern recognition*.

Carrera, D., Manganini, F., Boracchi, G., & Lanzarone, E. (2017). Defect detection in SEM images of nanofibrous materials. *IEEE Transactions on Industrial Informatics*, *13*(2), 551–561.

Chalapathy, R., Menon, A. K., & Chawla, S. (2018). Anomaly detection using one-class neural networks. arXiv:1802.06360.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 3213–3223).

Ehret, T., Davy, A., Morel, J. M., & Delbracio, M. (2019). Image anomalies: A review and synthesis of detection methods. *Journal of Mathematical Imaging and Vision*, *61*(5), 710–743.

Everingham, M., Eslami, S. M. A., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2015). The PASCAL visual object classes challenge: A retrospective. *International Journal of Computer Vision*, *111*(1), 98–136.

Gong, D., Liu, L., Le, V., Saha, B., Mansour, M.R., Venkatesh, S., Hengel, Avd. (2019). Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672–2680).

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning. Adaptive computation and machine learning series*. Cambridge, MA: MIT Press.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *IEEE Conference on computer vision and pattern recognition* (pp. 770–778).

Hendrycks, D., Basart, S., Mazeika, M., Mostajabi, M., Steinhardt, J., & Song, D. (2019a). A benchmark for anomaly segmentation. arXiv:1911.11132.

Hendrycks, D., Mazeika, M., Dietterich, T. (2019b). Deep anomaly detection with outlier exposure. In *International conference on learning representations*.

Huang, Y., Qiu, C., Guo, Y., Wang, X., & Yuan, K. (2018). Surface defect saliency of magnetic tile. In *2018 IEEE 14th international conference on automation science and engineering (CASE)* (pp. 612–617).

Kingma, D. P., & Welling, M. (2014). Auto-encoding variational Bayes. In *Proceedings of the international conference on learning representations (ICLR)*.

Krizhevsky, A., & Hinton, G. (2009). *Learning multiple layers of features from tiny images*. Technical report, University of Toronto.

Krizhevsky, A., Sutskever, I., Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Proceedings of the 25th international conference on neural information processing systems* (vol. 1, pp. 1097–1105).

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Lis, K., Nakka, K., Fua, P., & Salzmann, M. (2019). Detecting the unexpected via image resynthesis. In *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*.

Marchal, N., Moraldo, C., Blum, H., Siegwart, R., Cadena, C., & Gawel, A. (2020). Learning densities in feature space for reliable segmentation of indoor scenes. *IEEE Robotics and Automation Letters*, *5*(2), 1032–1038.

Nalisnick, E., Matsukawa, A., Teh, Y. W., Gorur, D., & Lakshminarayanan, B. (2019). Do deep generative models know what they don't know?

Napoletano, P., Piccoli, F., & Schettini, R. (2018). Anomaly detection in nanofibrous materials by CNN-based self-similarity. *Sensors*, *18*(1), 209.

Perera, P., & Patel, V. M. (2019). Learning deep features for one-class classification. *IEEE Transactions on Image Processing*, *28*(11), 5450–5463.

Pimentel, M. A., Clifton, D. A., Clifton, L., & Tarassenko, L. (2014). A review of novelty detection. *Signal Processing*, *99*, 215–249.

Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S. A., Binder, A., Müller, E., & Kloft, M. (2018). Deep one-class classification. In: Dy J, Krause A (eds) *Proceedings of the 35th international conference on machine learning, PMLR, proceedings of machine learning research* (vol. 80, pp. 4393–4402).

Sabokrou, M., Fayyaz, M., Fathy, M., Moayed, Z., & Klette, R. (2018). Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes. *Computer Vision and Image Understanding*, *172*, 88–97.

Saleh, B., Farahdi, A., & Elgammal, A. (2013). Object-centric anomaly detection by attribute-based reasoning. In *IEEE conference on computer vision and pattern recognition* (pp. 787–794).

Schlegl, T., Seeböck, P., Waldstein, S.M., Schmidt-Erfurth, U., & Langs, G. (2017). Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International conference on information processing in medical imaging* (pp. 146–157). Springer.

Schlegl, T., Seeböck, P., Waldstein, S., Langs, G., & Schmidt-Erfurth, U. (2019). f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks. In *Medical Image Analysis*, *54*.

Song, K., & Yan, Y. (2013). A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. *Applied Surface Science*, *285*, 858–864.

Steger, C. (2001). Similarity measures for occlusion, clutter, and illumination invariant object recognition. In: Radig B, Florczyk S (eds) Pattern recognition. Lecture notes in computer science. Springer, Berlin, vol. 2191, pp. 148–154.

Steger, C. (2002). Occlusion, clutter, and illumination invariant object recognition. *International Archives of Photogrammetry and Remote Sensing, vol XXXIV, part*, *3A*, 345–350.

Steger, C., Ulrich, M., & Wiedemann, C. (2018). *Machine vision algorithms and applications* (2nd ed.). Weinheim: Wiley-VCH.

Tukey, J. W. (1977). *Exploratory data analysis. Addison-Wesley series in behavioral science*. Reading, MA: Addison-Wesley.

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, *13*(4), 600–612.

Wieler, M., & Hahn, T. (2007). Weakly supervised learning for industrial optical inspection. In *29th Annual symposium of the German association for pattern recognition*. https://resources.mpi-inf.mpg.de/conference/dagm/2007/prizes.html.

Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., & Darrell, T. (2020). BDD100K: A diverse driving dataset for heterogeneous multitask learning. In *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 2633–2642).