EfficientAD: Accurate Visual Anomaly Detection at Millisecond-Level Latencies

Kilian Batzner[1] Lars Heckler[1,2] Rebecca König[1]
[1]MVTec Software GmbH, [2]Technical University of Munich
{kilian.batzner, lars.heckler, rebecca.koenig}@mvtec.com

Abstract

Detecting anomalies in images is an important task, especially in real-time computer vision applications. In this work, we focus on computational efficiency and propose a lightweight feature extractor that processes an image in less than a millisecond on a modern GPU. We then use a student–teacher approach to detect anomalous features. We train a student network to predict the extracted features of normal, i.e., anomaly-free training images. The detection of anomalies at test time is enabled by the student failing to predict their features. We propose a training loss that hinders the student from imitating the teacher feature extractor beyond the normal images. It allows us to drastically reduce the computational cost of the student–teacher model, while improving the detection of anomalous features. We furthermore address the detection of challenging logical anomalies that involve invalid combinations of normal local features, for example, a wrong ordering of objects. We detect these anomalies by efficiently incorporating an autoencoder that analyzes images globally. We evaluate our method, called EfficientAD, on 32 datasets from three industrial anomaly detection dataset collections. EfficientAD sets new standards for both the detection and the localization of anomalies. At a latency of two milliseconds and a throughput of six hundred images per second, it enables a fast handling of anomalies. Together with its low error rate, this makes it an economical solution for real-world applications and a fruitful basis for future research.

1. Introduction

In the past years, deep learning methods have continued to improve the state of the art across a wide range of computer vision applications. This progress has been accompanied by advances in making neural network architectures faster and more efficient [43, 59, 61, 63]. Modern classification architectures, for example, focus on characteristics such as latency, throughput, memory consumption, and the number of trainable parameters [32, 33, 54, 59, 60, 63]. This ensures that as networks become more capable, their computational requirements remain suitable for real-world applications. The field of visual anomaly detection has also seen rapid progress in the recent past, especially on industrial anomaly detection benchmarks [7, 9, 47, 50]. State-of-the-art anomaly detection methods, however, often sacrifice computational efficiency for an increased anomaly detection performance. Common techniques are ensembling, the use of large backbones, and increasing the input image resolution to up to 768×768 pixels.

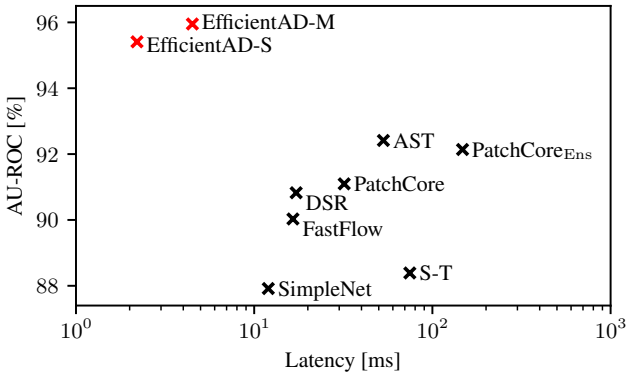

Figure 1. Anomaly detection performance vs. latency per image on an NVIDIA RTX A6000 GPU. Each AU-ROC value is an average of the image-level detection AU-ROC values on the MVTec AD [7, 9], VisA [69], and MVTec LOCO [8] dataset collections.

Real-world anomaly detection applications frequently put constraints on the computational requirements of a method. There are cases where detecting an anomaly too late can cause substantial economic damage, such as metal objects in a crop field entering the interior of a combine harvester. In other cases, even human health is at risk, for example, if a limb of a machine operator approaches a blade. Furthermore, industrial settings commonly involve strict runtime limits caused by high production rates [4]. Not adhering to these limits would decrease the production rate of the respective application and thus its economic viability. It is therefore essential to pay attention to the computational and economic cost of anomaly detection methods to keep them suitable for real-world applications.

In this work, we propose EfficientAD, a method that sets

new standards for both the anomaly detection performance and the inference runtime, as shown in Figure 1. We first introduce an efficient network architecture for computing expressive features in less than a millisecond on a modern GPU. To detect anomalous features, we use a student–teacher approach [10, 50, 62]. We train a student network to predict the features computed by a pretrained teacher network on normal, i.e., anomaly-free training images. Because the student is not trained on anomalous images, it generally fails to mimic the teacher on these. A large distance between the outputs of the teacher and the student thus enables the detection of anomalies at test time. To further increase this effect, Rudolph *et al*. [50] use *architectural* asymmetry between the teacher and the student. We instead propose *loss-induced* asymmetry in the form of a training loss that hinders the student from imitating the teacher beyond the normal images. This loss does not affect the computational cost at test time and does not restrict the architecture design. It allows us to use our efficient network architecture for both the student and the teacher, while improving the detection of anomalous features.

Identifying anomalous local features enables the detection of anomalies that are *structurally* different from the normal images, for example, contaminations or stains on manufactured products. A challenging problem, however, are violations of *logical* constraints regarding the position, size, arrangement, etc. of normal objects. To address this, EfficientAD includes an autoencoder that learns the logical constraints of training images and detects violations at test time. We show how to integrate the autoencoder efficiently with a student–teacher model. Furthermore, we present a method to improve the anomaly detection performance by calibrating the detection results of the autoencoder and the student–teacher model before combining their results.

Our contributions are summarized as follows:

- We substantially improve the state of the art for both the detection and the localization of anomalies on industrial benchmarks, at a latency of 2 ms and a throughput of more than 600 images per second.
- We propose an efficient network architecture to speed up feature extraction by an order of magnitude in comparison to the feature extractors used by recent methods [47, 50, 64].
- We introduce a training loss that significantly improves the anomaly detection performance of a student–teacher model without affecting its inference runtime.
- We achieve an efficient autoencoder-based detection of logical anomalies and propose a method for a calibrated combination of the detection results with those of a student–teacher model.

2. Related Work

2.1. Anomaly Detection Tasks

Visual anomaly detection is a rapidly growing area of research with a diverse range of applications, including medical imaging [3, 18, 36], autonomous driving [13, 23, 30], and industrial inspection [7, 17, 40]. Applications often have specific characteristics, such as the availability of image sequences in surveillance datasets [29, 35, 67] or the different modalities of medical imaging datasets (MRI [5], CT [3], X-ray [26], etc.). This work focuses on detecting anomalies in RGB or gray-scale images without conditioning the prediction on a sequence of images. We use industrial anomaly detection datasets to benchmark our proposed method against existing ones.

The introduction of the MVTec AD dataset [7, 9] has catalyzed the development of methods for industrial applications. It comprises 15 separate inspection scenarios, each consisting of a training set and a test set. Each training set contains only normal images, for example, defect-free screws, while the test sets also contain anomalous images. This represents a frequent challenge in real-world applications where the types and possible locations of defects are unknown during the development of the anomaly detection system. Therefore, it is a challenging yet crucial requirement that methods perform well when trained only on normal images.

Recently, several new industrial anomaly detection datasets have been introduced [8, 11, 25, 27, 37, 69]. The Visual Anomaly (VisA) dataset [69] and the MVTec Logical Constraints (MVTec LOCO) dataset [8] follow the design of MVTec AD and comprise twelve and five anomaly detection scenarios, respectively. They contain anomalies that are empirically more challenging than those of MVTec AD. Furthermore, MVTec LOCO contains not only structural anomalies, such as stains or scratches, but also logical anomalies. These are violations of logical constraints, for example, a wrong ordering or a wrong combination of normal objects. We refer to MVTec AD, VisA, and MVTec LOCO as dataset collections, as each scenario is a separate dataset consisting of a training and a test set. All three provide pixel-precise defect segmentation masks for evaluating the anomaly localization performance of a method.

2.2. Anomaly Detection Methods

Traditional computer vision algorithms have been applied successfully to industrial anomaly detection tasks for several decades [58]. These algorithms commonly fulfill the requirement of processing an image within a few milliseconds. Bergmann *et al*. [7] evaluate some of these methods and find that they fail when requirements such as well-aligned objects are not met. Deep-learning-based methods have been shown to handle such cases more robustly [7, 8].

A successful approach in the recent past has been to apply outlier detection and density estimation methods in the feature space of a pretrained and frozen convolutional neural network (CNN). If feature vectors can be mapped to input pixels, assigning their outlier scores to the respective pixels yields a 2D anomaly map of pixel anomaly scores. Common methods include multivariate Gaussian distributions [15, 28, 45], Gaussian Mixture Models [37, 68], Normalizing Flows [21, 44, 48, 49, 64], and the k-Nearest Neighbor (kNN) algorithm [14, 38, 39, 47]. A runtime bottleneck for kNN-based methods is the search for nearest neighbors during inference. With PatchCore [47], Roth *et al*. therefore perform kNN on a reduced database of clustered feature vectors. They achieve state-of-the-art anomaly detection results on MVTec AD. In our experiments, we include PatchCore and FastFlow [64], a recent Normalizing-Flow-based method with a comparatively low inference runtime.

Bergmann *et al*. [10] propose a student–teacher (S–T) framework for anomaly detection, in which the teacher is a pretrained frozen CNN. They train student networks to mimic the output of the teacher on the training images. Because the students have not seen anomalous images during training, they generally fail to predict the teacher's output on these images, which enables anomaly detection. Various modifications of S–T have been proposed [50, 53, 62]. Rudolph *et al*. [50] reach a competitive anomaly detection performance on MVTec AD by restricting the teacher to be an invertible neural network. We compare our method to their Asymmetric Student Teacher (AST) approach and to the original S–T method [10].

Generative models such as autoencoders [6, 8, 12, 19, 31, 41, 52] and GANs [2, 20, 42, 55, 56] have been used extensively for anomaly detection. Recent autoencoder-based methods rely on accurate reconstructions of normal images and inaccurate reconstructions of anomalous images [8, 12, 19, 41]. This enables detecting anomalies by comparing the reconstruction to the input image. A common problem are false-positive detections caused by inaccurate reconstructions of normal images, e.g., blurry reconstructions. To avoid this, GCAD [8] lets an autoencoder reconstruct images in the feature space of a pretrained network. Another recent reconstruction-based method is DSR [66], which uses the latent space of a pretrained autoencoder and generates synthetic anomalies in it. Similarly, the recently proposed SimpleNet [34] generates synthetic anomalies in a pretrained feature space to train a discriminator network for detecting anomalous features. In our experiments, we include GCAD, DSR, and SimpleNet.

3. Method

We describe the components of EfficientAD in the following subsections. It begins with the efficient extraction of features from a pretrained neural network in Sec. 3.1. We detect anomalous features at test time using a lightweight student–teacher model, as described in Sec. 3.2. A key challenge is to achieve a competitive anomaly detection performance while keeping the overall runtime low. To this end, we introduce a novel loss function for the training of a student–teacher model. In Sec. 3.3, we explain how to efficiently detect logical anomalies with an autoencoder-based approach. Finally, we provide a solution for calibrating and combining the detection results of the autoencoder with those of the student–teacher model in Sec. 3.4.

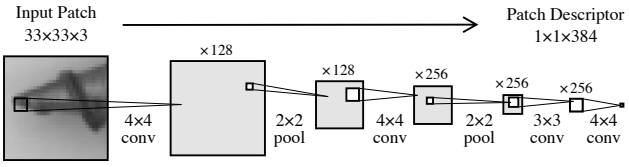

Figure 2. Patch description network (PDN) architecture of EfficientAD-S. Applying it to an image in a fully convolutional manner yields all features in a single forward pass.

3.1. Efficient Patch Descriptors

Recent anomaly detection methods commonly use the features of a deep pretrained network, such as a WideResNet-101 [47, 65]. We use a network with a drastically reduced depth as a feature extractor. It consists of only four convolutional layers and is visualized in Figure 2. Each output neuron has a receptive field of 33×33 pixels and thus each output feature vector describes a 33×33 patch. Due to this clear correspondence, we refer to the network as a patch description network (PDN). The PDN is fully convolutional and can be applied to an image of variable size to generate all feature vectors in a single forward pass.

The S–T method [10] also uses features from networks with only few convolutional layers. The computational cost of these networks is nevertheless high because of the lack of downsampling in convolutional and pooling layers. The number of parameters of the networks used by S–T is comparably low (between 1.6 and 2.7 million per network). Yet, executing a single network takes longer and requires more memory in our experiments than a U-Net [46] with 31 million parameters, an architecture used by the GCAD method [8]. This demonstrates how the number of parameters can be a misleading proxy metric for the latency, throughput, and memory footprint of a method. Modern classification architectures typically perform downsampling early to reduce the size of feature maps and thus the runtime and memory requirements [22]. We implement this in our PDN via strided average-pooling layers after the first and the second convolutional layer. With the proposed PDN, we are able to obtain the features for an image of size 256×256 in less than 800 µs on an NVIDIA RTX A6000 GPU.

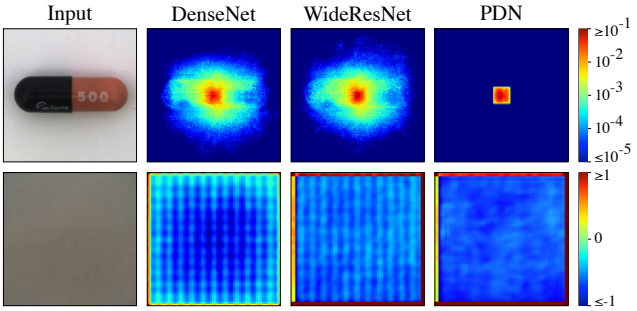

Figure 3. Upper row: absolute gradient of a single feature vector, located in the center of the output, with respect to each input pixel, averaged across input and output channels. Lower row: Average feature map of the first output channel across 1000 randomly chosen images from ImageNet [51]. The mean of these images is shown on the left. The feature maps of the DenseNet [24] and the WideResNet exhibit strong artifacts.

To make the PDN generate expressive features, we distill a deep pretrained classification network into it. For a controlled comparison, we use the same pretrained features as PatchCore [47] from a WideResNet-101. We train the PDN on images from ImageNet [51] by minimizing the mean squared difference between its output and the features extracted from the pretrained network. We provide the full list of training hyperparameters in the supplementary material. Besides higher efficiency, the PDN has another benefit in comparison to the deep networks used by recent methods. By design, a feature vector generated by the PDN only depends on the pixels in its respective 33×33 patch. The feature vectors of pretrained classifiers, on the other hand, exhibit long-range dependencies on other parts of the image. This is shown in Figure 3, using PatchCore's feature extractors as an example. The well-defined receptive field of the PDN ensures that an anomaly in one part of the image cannot trigger anomalous feature vectors in other, distant parts, which would impair the localization of anomalies.

3.2. Lightweight Student–Teacher

For detecting anomalous feature vectors, we use a student–teacher (S–T) approach in which the teacher is given by our distilled PDN. Since we can execute the PDN in under a millisecond, we use its architecture for the student as well, resulting in a low overall latency. This lightweight student–teacher pair, however, lacks techniques used by previous methods to increase the anomaly detection performance: ensembling multiple teachers and students [10], using features from a pyramid of layers [62], and using architectural asymmetry between the student and the teacher network [50]. We therefore introduce a training loss that substantially improves the detection of anomalies without affecting the computational requirements at test time.

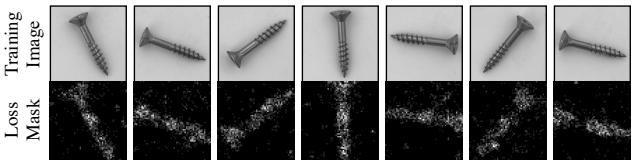

Figure 4. Randomly picked loss masks generated by the hard feature loss during training. The brightness of a mask pixel indicates how many of the dimensions of the respective feature vector were selected for backpropagation. The student network already mimics the teacher well on the background and thus focuses on learning the features of differently rotated screws.

We observe that in the standard S–T framework, increasing the number of training images can improve the student's ability to imitate the teacher on anomalies. This worsens the anomaly detection performance. At the same time, deliberately decreasing the number of training images can suppress important information about normal images. Our goal is to show the student enough data so that it can mimic the teacher sufficiently on normal images while avoiding generalization to anomalous images. Similar to Online Hard Example Mining [57], we therefore restrict the student's loss to the most relevant parts of an image. These are the patches where the student currently mimics the teacher the least. We propose a hard feature loss, which only uses the output elements with the highest loss for backpropagation.

Formally, we apply a teacher T and a student S to a training image I, which yields $T(I) \in \mathbb{R}^{C\times W\times H}$ and $S(I) \in \mathbb{R}^{C\times W\times H}$. We compute the squared difference for each tuple (c, w, h) as $D_{c,w,h} = (T(I)_{c,w,h} - S(I)_{c,w,h})^2$. Based on a mining factor $p_{\text{hard}} \in [0, 1]$, we then compute the p_{hard}-quantile of the elements of D. Given the p_{hard}-quantile d_{hard}, we compute the training loss L_{hard} as the mean of all $D_{c,w,h} \geq d_{\text{hard}}$. Setting p_{hard} to zero would yield the original S–T loss. In our experiments, we set p_{hard} to 0.999, which corresponds to using, on average, ten percent of the values in each of the three dimensions of D for backpropagation. Figure 4 visualizes the effect of the hard feature loss for $p_{\text{hard}} = 0.999$. During inference, the 2D anomaly score map $M \in \mathbb{R}^{W\times H}$ is given by $M_{w,h} = C^{-1}\sum_c D_{c,w,h}$, i.e., by D averaged across channels. It assigns an anomaly score to each feature vector.

In addition to the hard feature loss, we use a loss penalty during training that further hinders the student from imitating the teacher on images that are not part of the normal training images. In the standard S–T framework, the teacher is pretrained on an image classification dataset, or it is a distilled version of such a pretrained network. The student is not trained on that pretraining dataset but only on the application's normal images. We propose to also use the images from the teacher's pretraining during the train-

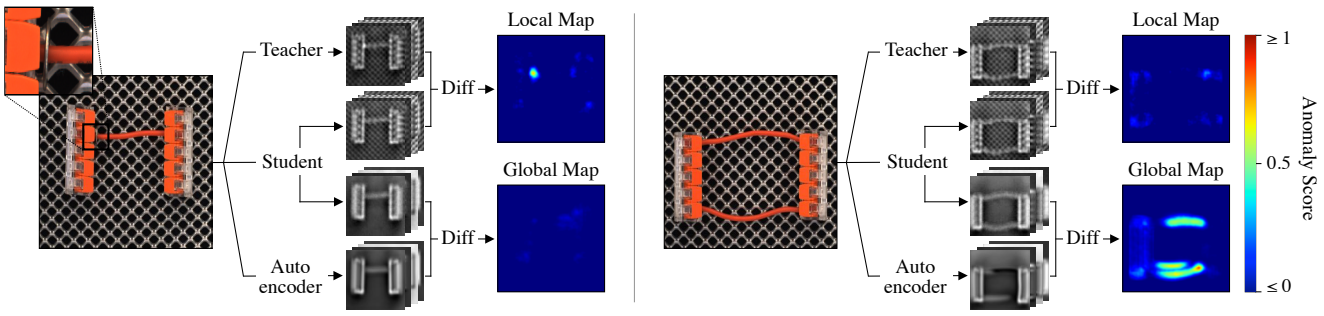

Figure 5. EfficientAD applied to two test images from MVTec LOCO. Normal input images contain a horizontal cable connecting the two splicing connectors at an arbitrary height. The anomaly on the left is a foreign object in the form of a small metal washer at the end of the cable. It is visible in the local anomaly map because the outputs of the student and the teacher differ. The logical anomaly on the right is the presence of a second cable. The autoencoder fails to reconstruct the two cables on the right in the feature space of the teacher. The student also predicts the output of the autoencoder in addition to that of the teacher. Because its receptive field is restricted to small patches of the image, it is not influenced by the presence of the additional red cable. This causes the outputs of the autoencoder and the student to differ. "Diff" refers to computing the element-wise squared difference between two collections of output feature maps and computing its average across feature maps. To obtain pixel anomaly scores, the anomaly maps are resized to match the input image using bilinear interpolation.

ing of the student. Specifically, we sample a random image P from the pretraining dataset, in our case ImageNet, in each training step. We compute the loss of the student as $L_{\text{ST}} = L_{\text{hard}} + (CWH)^{-1} \sum_c \|S(P)_c\|_F^2$. This penalty hinders the student from generalizing its imitation of the teacher to out-of-distribution images.

3.3. Logical Anomaly Detection

There are many types of logical anomalies, such as missing, misplaced, or surplus objects or the violation of geometrical constraints, for example, the length of a screw. As recommended by the authors of the MVTec LOCO dataset [8], we use an autoencoder for learning logical constraints of the training images and detecting violations of these constraints. Figure 5 depicts the anomaly detection methodology for EfficientAD. It consists of the aforementioned student–teacher pair and an autoencoder. The autoencoder is trained to predict the output of the teacher. Formally, we apply an autoencoder A to a training image I, yielding $A(I) \in \mathbb{R}^{C \times W \times H}$, and compute the loss as $L_{\text{AE}} = (CWH)^{-1} \sum_c \|T(I)_c - A(I)_c\|_F^2$. We use a standard convolutional autoencoder comprising strided convolutions in the encoder and bilinear upsampling in the decoder. We provide the detailed hyperparameters of its layers in the supplementary material.

In contrast to the patch-based student, the autoencoder must encode and decode the complete image through a bottleneck of 64 latent dimensions. On images with logical anomalies, the autoencoder usually fails to generate the correct latent code for reconstructing the image in the teacher's feature space. However, its reconstructions are also flawed on normal images, as autoencoders generally struggle with reconstructing fine-grained patterns [12, 16]. This is the case for the background grids in Figure 5. Using the difference between the teacher's output and the autoencoder's reconstruction as an anomaly map would cause false-positive detections in these cases. Instead, we double the number of output channels of our student network and train it to predict the output of the autoencoder in addition to the output of the teacher. Let $S'(I) \in \mathbb{R}^{C \times W \times H}$ denote the additional output channels of the student. The student's additional loss is then $L_{\text{STAE}} = (CWH)^{-1} \sum_c \|A(I)_c - S'(I)_c\|_F^2$.

The student learns the systematic reconstruction errors of the autoencoder on normal images, e.g., blurry reconstructions. At the same time, it does not learn the reconstruction errors for anomalies because these are not part of the training set. This makes the difference between the autoencoder's output and the student's output well-suited for computing the anomaly map. Analogous to the student–teacher pair, the anomaly map is the squared difference between the two outputs, averaged across channels. We refer to this anomaly map as the global anomaly map and to the anomaly map generated by the student–teacher pair as the local anomaly map. We average these two anomaly maps to compute the combined anomaly map and use its maximum value as the image-level anomaly score. The combined anomaly map thus contains the detection results of the student–teacher pair and the detection results of the autoencoder–student pair. Sharing the student's hidden layers in the computation of these detection results allows our method to maintain low computational requirements, while enabling the detection of structural and logical anomalies.

3.4. Anomaly Map Normalization

The local and the global anomaly map must be normalized to similar scales before averaging them to obtain the

combined anomaly map. This is important for cases where the anomaly is only detected in one of the maps, such as in Figure 5. Otherwise, noise in one map could make accurate detections in the other map indiscernible in the combined map. To estimate the scale of the noise in normal images, we use validation images, i.e., unseen images from the training set. For each of the two anomaly map types, we compute the set of all pixel anomaly scores across the validation images. We then compute two p-quantiles for each set: q_a and q_b, for $p = a$ and $p = b$, respectively. We determine a linear transformation that maps q_a to an anomaly score of 0 and q_b to a score of 0.1. At test time, the local and global anomaly maps are normalized with the respective linear transformation.

By using quantiles, the normalization becomes robust to the distribution of anomaly scores on normal images, which can vary between scenarios. Whether the scores between q_a and q_b are normally distributed or a mixture of Gaussians or follow another distribution has no influence on the normalization. Our experiments include an ablation study on the values of a and b. The choice of the mapping destination values 0 and 0.1 has no effect on anomaly detection metrics such as the area under the ROC curve (AU-ROC). That is because the AU-ROC only depends on the ranking of scores, not on their scale. We choose 0 and 0.1 because they yield maps that are suitable for a standard zero-to-one color scale.

4. Experiments

We compare EfficientAD to AST [50], DSR [66], FastFlow [64], GCAD [8], PatchCore [47], SimpleNet [34], and S–T [10], using official implementations where available. We provide configuration details for all evaluated methods in the supplementary material. GCAD consists of an ensemble of two anomaly detection models that use different feature extractors. We find that one of the two ensemble members performs better on average than the combined ensemble and therefore report the results for this member. This reduces the latency reported for GCAD by a factor of two. For SimpleNet, we are able to reproduce the official results but find that SimpleNet tunes the training duration on the test images of a scenario. During training, the model is repeatedly evaluated on all test images and the maximum of all obtained test scores is reported after training. We disable this technique, since it overestimates the actual performance of the model on unseen images. In practice, it would furthermore require a validation set with anomalous images. MVTec AD, VisA, and MVTec LOCO do not include anomalous images in their training and validation sets to avoid defect-type-specific tuning of hyperparameters. For SimpleNet, we therefore evaluate the final trained model, following common practice.

For PatchCore, we include two variants: the default single model variant, for which the authors report the lowest latency, and the ensemble variant, denoted by $\text{PatchCore}_{\text{Ens}}$. We are able to reproduce the official results but disable the cropping of the center 76.6 % of input images for a fair comparison. In the case of MVTec AD, 99.9 % of the defects lie fully or partially within this cropped area. In real-world applications, anomalies can occur outside of this area as well. We disable custom cropping, as it implies knowledge about the anomalies in the test set. For FastFlow, we use the version based on the WideResNet-50-2 feature extractor, as it is similar to the WideResNet used by PatchCore, SimpleNet, and our method. We use the implementation provided by the Intel anomalib [1] but disable early stopping, i.e., the scenario-specific tuning of the training duration on test images, analogously to SimpleNet. With early stopping enabled, EfficientAD itself achieves an image-level detection AU-ROC of 99.8 % on MVTec AD.

For our method, we evaluate two variants: EfficientAD-S and EfficientAD-M. EfficientAD-S uses the architecture displayed in Figure 2 for the teacher and the student. For EfficientAD-M, we double the number of kernels in the hidden convolutional layers of the teacher and the student. Furthermore, we insert a 1×1 convolution after the second pooling layer and after the last convolutional layer. We provide a list of implementation details, such as the learning rate schedule, in the supplementary material.

We evaluate each method on the 32 anomaly detection scenarios of MVTec AD, VisA, and MVTec LOCO. The anomaly detection performance of a method is measured with the AU-ROC based on its predicted image-level anomaly scores. We measure the anomaly localization performance using the AU-PRO segmentation metric up to a false positive rate of 30 %, as recommended by [7]. For MVTec LOCO, we use the AU-sPRO metric [8], a generalization of the AU-PRO metric for evaluating the localization of logical anomalies. The supplementary material provides the results for additional anomaly detection metrics, such as the area under the precision-recall curve and the pixel-wise AU-ROC.

When reporting the AU-ROC or AU-PRO for a dataset collection, we follow the policy of the dataset authors. For each collection, we evaluate the respective metric for each scenario and then compute the mean across scenarios. For MVTec LOCO, we use the official evaluation script, which gives logical and structural anomalies an equal weight in the computed metrics. When reporting the average AU-ROC or AU-PRO on the three dataset collections, we compute the average of the three dataset means. Thus, an overall average score weights logical anomalies and structural anomalies by roughly one-sixth and five-sixths, respectively. We provide the evaluation results for each of the 32 anomaly detection scenarios individually in the supplementary material to enable an evaluation with a custom weighting.

Method	Detect. AU-ROC	Segment. AU-PRO	Latency [ms]	Throughput [img / s]
GCAD	85.4	88.0	11	121
SimpleNet	87.9	74.4	12	194
S–T	88.4	89.7	75	16
FastFlow	90.0	86.5	17	120
DSR	90.8	78.6	17	104
PatchCore	91.1	80.9	32	76
$\text{PatchCore}_{\text{Ens}}$	92.1	80.7	148	13
AST	92.4	77.2	53	41
EfficientAD-S	95.4 (± 0.06)	92.5 (± 0.05)	**2.2** (± 0.01)	**614** (± 2)
EfficientAD-M	**96.0** (± 0.09)	**93.3** (± 0.04)	4.5 (± 0.01)	269 (± 1)

Table 1. Anomaly detection and anomaly localization performance in comparison to the latency and throughput. Each AU-ROC and AU-PRO percentage is an average of the mean AU-ROCs and mean AU-PROs, respectively, on MVTec AD, VisA, and MVTec LOCO. For EfficientAD, we report the mean and standard deviation of five runs.

Method	MAD	LOCO	VisA	Mean	LOCO Logic.	LOCO Struct.
GCAD	89.1	83.3	83.7	85.4	83.9	82.7
SimpleNet	98.2	77.6	87.9	87.9	71.5	83.7
S–T	93.2	77.4	94.6	88.4	66.5	88.3
FastFlow	96.9	79.2	93.9	90.0	75.5	82.9
DSR	98.1	82.6	91.8	90.8	75.0	90.2
PatchCore	98.7	80.3	94.3	91.1	75.8	84.8
$\text{PatchCore}_{\text{Ens}}$	**99.3**	79.4	97.7	92.1	71.0	87.7
AST	98.9	83.4	94.9	92.4	79.7	87.1
EfficientAD-S	98.8	90.0	97.5	95.4	85.8	94.1
EfficientAD-M	99.1	**90.7**	**98.1**	**96.0**	**86.8**	**94.7**

Table 2. Mean anomaly detection AU-ROC percentages per dataset collection (left) and on the logical and structural anomalies of MVTec LOCO (right). For EfficientAD, we report the mean of five runs. Performing method development solely on MVTec AD (MAD) becomes prone to overfitting design choices to the few remaining misclassified test images.

a (for q_a)	0.5	0.8	**0.9**	0.95	0.98	0.99
AU-ROC	95.9	95.9	96.0	95.9	95.9	95.8
b (for q_b)	0.95	0.98	0.99	**0.995**	0.998	0.999
AU-ROC	95.8	95.9	96.0	96.0	95.9	95.9
p_{hard}	0	0.9	0.99	**0.999**	0.9999	0.99999
AU-ROC	94.9	94.9	95.7	96.0	95.8	95.7

Table 3. Mean anomaly detection AU-ROC of EfficientAD-M on MVTec AD, VisA, and MVTec LOCO when varying the locations of quantiles. These are the two sampling points a and b of the quantile-based map normalization and the mining factor p_{hard}. Setting p_{hard} to zero disables the proposed hard feature loss. Default values used in our experiments are highlighted in bold.

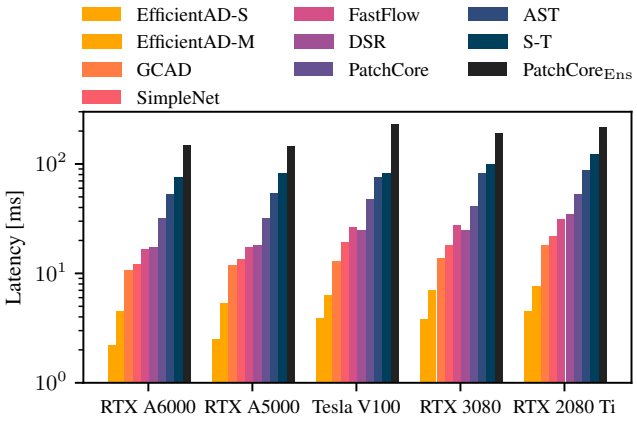

Figure 6. Latency per GPU. The ranking of methods is the same on each GPU, except for two cases in which DSR is slightly faster than FastFlow.

Table 1 reports the overall anomaly detection performance for each method. EfficientAD achieves a strong image-level detection and pixel-level localization of anomalies. Reliably localizing anomalies in an image provides explainable detection results and allows the discovery of spurious correlations in detections. It also enables a flexible postprocessing, such as excluding defect segmentations based on their size.

Table 2 breaks down the overall anomaly detection performance into the three dataset collections. It shows that the lead of EfficientAD on MVTec LOCO is in equal parts due to its performance on logical and on structural anomalies. In Table 3, we assess the robustness of EfficientAD to varying hyperparameters.

Furthermore, we measure the computational cost of each method during inference. As explained above, the number of parameters can be a misleading proxy metric for the latency and throughput of convolutional architectures since it does not consider the resolution of a convolution's input feature map, i.e., how often a parameter is used in a forward pass. Similarly, the number of floating point operations (FLOPs) can be misleading since it does not take into account how easily computations can be parallelized. For transparency, we report the number of parameters, the number of FLOPs, and the memory footprint of each method in the supplementary material. Here, we focus on the metrics that are most relevant in anomaly detection applications: the latency and the throughput. We measure the latency with a batch size of 1 and the throughput with a batch size of 16. Table 1 reports the measurements for each method on an NVIDIA RTX A6000 GPU. Figure 6 shows the latency of each method on each of the GPUs in our experimental setup. The supplementary material contains a detailed description of our timing methodology.

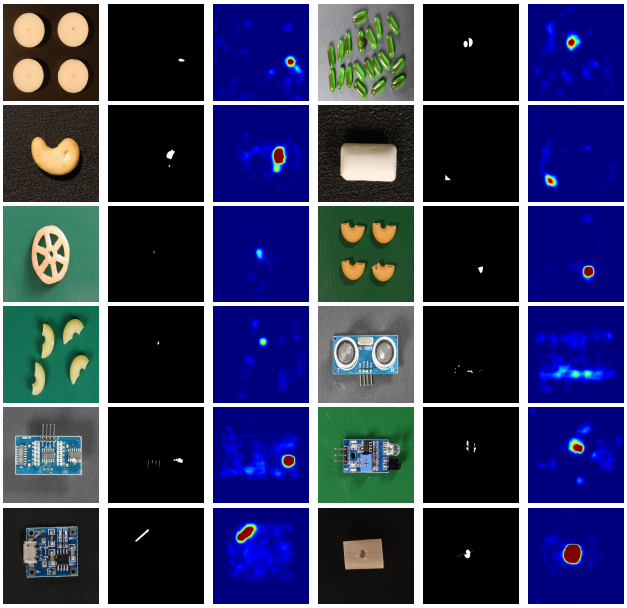

Figure 7. Non-cherry-picked qualitative results of EfficientAD on VisA. For each of its 12 scenarios, we show a randomly sampled defect image, the ground truth segmentation mask, and the anomaly map generated by EfficientAD-M.

In Figure 7, we show randomly sampled qualitative results of EfficientAD on the VisA dataset collection. The supplementary material provides qualitative results for the other evaluated methods and dataset collections as well.

We examine the effects of the components of EfficientAD in the ablation study shown in Table 4 and Table 5. For experiments without the proposed quantile-based map normalization, we use a Gaussian-based map normalization as a baseline instead. There, we compute the linear transformation parameters such that pixel anomaly scores on the validation set have a mean of zero and a variance of one. This baseline normalization is sensitive to the distribution of validation anomaly scores, which can vary between scenarios. The quantile-based normalization is independent of how the scores between q_a and q_b are distributed and performs substantially better than the baseline.

We also evaluate the effect of the two proposed loss terms for training the student–teacher pair. The hard feature loss increases the anomaly detection AU-ROC by 1.0 % in Table 4. This improvement alone is greater than or equal to each of the improvement margins between the consecutive rows of FastFlow, DSR, PatchCore, PatchCore$_{\text{Ens}}$, and AST in Table 1. The student's penalty on pretraining images further improves the anomaly detection performance. Notably, the proposed map normalization, the hard feature loss, and the pretraining penalty keep the computational requirements of EfficientAD low, while creating a substantial margin w.r.t. the anomaly detection performance.

	Detection AU-ROC	Diff.	Latency [ms]
PDN	93.2		2.2
↪ with map normalization	94.0	+ 0.8	2.2
↪ with hard feature loss	95.0	+ 1.0	2.2
↪ with pretraining penalty	95.4	+ 0.4	2.2
EfficientAD-S	95.4		2.2
EfficientAD-M	96.0	+ 0.6	4.5

Table 4. Cumulative ablation study in which techniques are gradually combined to form EfficientAD. Each AU-ROC percentage is an average of the mean AU-ROCs on MVTec AD, VisA, and MVTec LOCO.

	Detection AU-ROC	Diff.	Latency [ms]
EfficientAD-S	95.4		2.2
Without map normalization	94.7	- 0.7	2.2
Without hard feature loss	94.7	- 0.7	2.2
Without pretraining penalty	95.0	- 0.4	2.2

Table 5. Isolated ablation study in which techniques are separately removed from EfficientAD-S.

5. Conclusion

In this paper, we introduce EfficientAD, a method with a strong anomaly detection performance and a high computational efficiency. It sets new standards for the detection of structural as well as logical anomalies. Both EfficientAD-S and EfficientAD-M outperform other methods on the detection and the localization of anomalies by a large margin. Compared to AST, the second-best method, EfficientAD-S reduces the latency by a factor of 24 and increases the throughput by a factor of 15. Its low latency, high throughput, and high detection rate make it suitable for real-world applications. For future anomaly detection research, EfficientAD is an important baseline and a fruitful foundation. Its efficient patch description network, for instance, can be used as a feature extractor in other anomaly detection methods as well to reduce their latency.

Limitations. The student–teacher model and the autoencoder are designed to detect anomalies of different types. The autoencoder detects logical anomalies, while the student–teacher model detects coarse and fine-grained structural anomalies. Fine-grained logical anomalies, however, remain a challenge – for example a screw that is two millimeters too long. To detect these, practitioners would have to use traditional metrology methods [58]. As for the limitations in comparison to other recent anomaly detection methods: In contrast to kNN-based methods, our approach requires training, especially for the autoencoder to learn the logical constraints of normal images. This takes twenty minutes in our experimental setup.

References

[1] Samet Akcay, Dick Ameln, Ashwin Vaidya, Barath Lakshmanan, Nilesh Ahuja, and Utku Genc. Anomalib: A deep learning library for anomaly detection. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 1706–1710. IEEE, 2022. 6

[2] Samet Akcay, Amir Atapour-Abarghouei, and Toby P Breckon. Ganomaly: Semi-supervised anomaly detection via adversarial training. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14*, pages 622–637. Springer, 2019. 3

[3] Samuel G Armato III, Geoffrey McLennan, Luc Bidaut, Michael F McNitt-Gray, Charles R Meyer, Anthony P Reeves, Binsheng Zhao, Denise R Aberle, Claudia I Henschke, Eric A Hoffman, et al. The lung image database consortium (lidc) and image database resource initiative (idri): a completed reference database of lung nodules on ct scans. *Medical physics*, 38(2):915–931, 2011. 2

[4] Donald Bailey. *Implementing Machine Vision Systems Using FPGAs*, pages 1103–1136 in "Machine Vision Handbook" by Bruce G. Batchelor. Springer London, London, 2012. 1

[5] Spyridon Bakas, Hamed Akbari, Aristeidis Sotiras, Michel Bilello, Martin Rozycki, Justin S. Kirby, et al. Advancing the cancer genome atlas glioma MRI collections with expert segmentation labels and radiomic features. *Scientific Data*, 4(1), 2017. 2

[6] Christoph Baur, Benedikt Wiestler, Shadi Albarqouni, and Nassir Navab. Deep autoencoding models for unsupervised anomaly segmentation in brain mr images. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pages 161–169. Springer International Publishing, 2019. 3

[7] Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, and Carsten Steger. The MVTec Anomaly Detection Dataset: A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. *International Journal of Computer Vision*, 129(4):1038–1059, 2021. 1, 2, 6

[8] Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, and Carsten Steger. Beyond Dents and Scratches: Logical Constraints in Unsupervised Anomaly Detection and Localization. *International Journal of Computer Vision*, 130(4):947—-969, 2022. 1, 2, 3, 5, 6

[9] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. MVTec AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9584–9592, 2019. 1, 2

[10] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Uninformed Students: Student-Teacher Anomaly Detection With Discriminative Latent Embeddings. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4182–4191, 2020. 2, 3, 4, 6

[11] Paul Bergmann, Xin Jin, David Sattlegger, and Carsten Steger. The MVTec 3D-AD Dataset for Unsupervised 3D Anomaly Detection and Localization. In *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP*, pages 202–213. INSTICC, SciTePress, 2022. 2

[12] Paul Bergmann, Sindy Löwe, Michael Fauser, David Sattlegger, and Carsten Steger. Improving Unsupervised Defect Segmentation by Applying Structural Similarity to Autoencoders. In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP*, pages 372–380. INSTICC, SciTePress, 2019. 3, 5

[13] Hermann Blum, Paul-Edouard Sarlin, Juan Nieto, Roland Siegwart, and Cesar Cadena. Fishyscapes: A Benchmark for Safe Semantic Segmentation in Autonomous Driving. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 2403–2412, 2019. 2

[14] Niv Cohen and Yedid Hoshen. Sub-image anomaly detection with deep pyramid correspondences. *arXiv preprint arXiv:2005.02357v1*, 2020. 3

[15] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. PaDiM: A Patch Distribution Modeling Framework for Anomaly Detection and Localization. In *Pattern Recognition. ICPR International Workshops and Challenges*, pages 475–489. Springer International Publishing, 2021. 3

[16] Alexey Dosovitskiy and Thomas Brox. Generating Images with Perceptual Similarity Metrics based on Deep Networks. In *Advances in Neural Information Processing Systems*, pages 658–666, 2016. 5

[17] Thibaud Ehret, Axel Davy, Jean-Michel Morel, and Mauricio Delbracio. Image Anomalies: A Review and Synthesis of Detection Methods. *Journal of Mathematical Imaging and Vision*, 61(5):710–743, 2019. 2

[18] Tharindu Fernando, Harshala Gammulle, Simon Denman, Sridha Sridharan, and Clinton Fookes. Deep learning for medical anomaly detection – a survey. *ACM Computing Surveys*, 54(7), 2021. 2

[19] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton Van Den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1705–1714, 2019. 3

[20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 3

[21] Denis Gudovskiy, Shun Ishizaka, and Kazuki Kozuka. CFLOW-AD: Real-Time Unsupervised Anomaly Detection With Localization via Conditional Normalizing Flows. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 98–107, 2022. 3

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 3

[23] Dan Hendrycks, Steven Basart, Mantas Mazeika, Andy Zou, Joseph Kwon, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 8759–8773. PMLR, 17–23 Jul 2022. 2

[24] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 4

[25] Yibin Huang, Congying Qiu, Yue Guo, Xiaonan Wang, and Kui Yuan. Surface defect saliency of magnetic tile. In *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*, pages 612–617, 2018. 2

[26] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 590–597, 2019. 2

[27] Stepan Jezek, Martin Jonak, Radim Burget, Pavel Dvorak, and Milos Skotak. Deep learning-based defect detection of metal parts: evaluating current methods in complex conditions. In *2021 13th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pages 66–71. IEEE, 2021. 2

[28] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9664–9674, 2021. 3

[29] Wei-Xin Li, Vijay Mahadevan, and Nuno Vasconcelos. Anomaly Detection and Localization in Crowded Scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(1):18–32, 2013. 2

[30] Krzysztof Lis, Krishna Kanth Nakka, Pascal Fua, and Mathieu Salzmann. Detecting the unexpected via image resynthesis. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2152–2161, 2019. 2

[31] Wenqian Liu, Runze Li, Meng Zheng, Srikrishna Karanam, Ziyan Wu, Bir Bhanu, Richard J. Radke, and Octavia Camps. Towards visually explaining variational autoencoders. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8639–8648, 2020. 3

[32] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 1

[33] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022. 1

[34] Zhikang Liu, Yiming Zhou, Yuansheng Xu, and Zilei Wang. Simplenet: A simple network for image anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20402–20411, June 2023. 3, 6

[35] Cewu Lu, Jianping Shi, and Jiaya Jia. Abnormal Event Detection at 150 FPS in MATLAB. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2720–2727, 2013. 2

[36] Bjoern H. Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, et al. The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). *IEEE Transactions on Medical Imaging*, 34(10):1993–2024, 2015. 2

[37] Pankaj Mishra, Riccardo Verk, Daniele Fornasier, Claudio Piciarelli, and Gian Luca Foresti. Vt-adl: A vision transformer network for image anomaly detection and localization. In *2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)*, pages 01–06. IEEE, 2021. 2, 3

[38] Paolo Napoletano, Flavio Piccoli, and Raimondo Schettini. Anomaly Detection in Nanofibrous Materials by CNN-Based Self-Similarity. *Sensors*, 18(1):209, 2018. 3

[39] Tiago S Nazare, Rodrigo F de Mello, and Moacir A Ponti. Are pre-trained cnns good feature extractors for anomaly detection in surveillance videos? *arXiv preprint arXiv:1811.08495v1*, 2018. 3

[40] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. Deep learning for anomaly detection: A review. *ACM Comput. Surv.*, 54(2), mar 2021. 2

[41] Hyunjong Park, Jongyoun Noh, and Bumsub Ham. Learning memory-guided normality for anomaly detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14360–14369, 2020. 3

[42] Pramuditha Perera, Ramesh Nallapati, and Bing Xiang. Ocgan: One-class novelty detection using gans with constrained latent representations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2893–2901, 2019. 3

[43] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1

[44] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015. 3

[45] Oliver Rippel., Arnav Chavan., Chucai Lei., and Dorit Merhof. Transfer learning gaussian anomaly detection by fine-tuning representations. In *Proceedings of the 2nd International Conference on Image Processing and Vision Engineering - IMPROVE,*, pages 45–56. INSTICC, SciTePress, 2022. 3

[46] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241. Springer International Publishing, 2015. 3

[47] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14318–14328, 2022. 1, 2, 3, 4, 6

[48] Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn. Same same but differnet: Semi-supervised defect detection with normalizing flows. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1906–1915, 2021. 3

[49] Marco Rudolph, Tom Wehrbein, Bodo Rosenhahn, and Bastian Wandt. Fully convolutional cross-scale-flows for image-based defect detection. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1829–1838, 2022. 3

[50] Marco Rudolph, Tom Wehrbein, Bodo Rosenhahn, and Bastian Wandt. Asymmetric student-teacher networks for industrial anomaly detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2592–2602, 2023. 1, 2, 3, 4, 6

[51] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 4

[52] Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, MLSDA'14, page 4–11, New York, NY, USA, 2014. Association for Computing Machinery. 3

[53] Mohammadreza Salehi, Niousha Sadjadi, Soroosh Baselizadeh, Mohammad H. Rohban, and Hamid R. Rabiee. Multiresolution knowledge distillation for anomaly detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14897–14907, 2021. 3

[54] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 1

[55] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery. In *International Conference on Information Processing in Medical Imaging*, pages 146–157. Springer, 2017. 3

[56] Thomas Schlegl, Philipp Seeböck, Sebastian Waldstein, Georg Langs, and Ursula Schmidt-Erfurth. f-AnoGAN: Fast Unsupervised Anomaly Detection with Generative Adversarial Networks. *Medical Image Analysis*, 54, 2019. 3

[57] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 761–769, 2016. 4

[58] Carsten Steger, Markus Ulrich, and Christian Wiedemann. *Machine Vision Algorithms and Applications*. Wiley-VCH, Weinheim, 2nd edition, 2018. 2, 8

[59] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 1

[60] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International conference on machine learning*, pages 10096–10106. PMLR, 2021. 1

[61] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020. 1

[62] Guodong Wang, Shumin Han, Errui Ding, and Di Huang. Student-teacher feature pyramid matching for anomaly detection. In *The British Machine Vision Conference (BMVC)*, 2021. 2, 3, 4

[63] Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10809–10818, 2022. 1

[64] Jiawei Yu, Ye Zheng, Xiang Wang, Wei Li, Yushuang Wu, Rui Zhao, and Liwei Wu. Fastflow: Unsupervised anomaly detection and localization via 2d normalizing flows. *arXiv preprint arXiv:2111.07677v1*, 2021. 2, 3, 6

[65] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Edwin R. Hancock Richard C. Wilson and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 87.1–87.12. BMVA Press, September 2016. 3

[66] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. Dsr–a dual subspace re-projection network for surface anomaly detection. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXI*, pages 539–554. Springer, 2022. 3, 6

[67] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 589–597, 2016. 2

[68] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International conference on learning representations*, 2018. 3

[69] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXX*, pages 392–408. Springer, 2022. 1, 2