# Generation of Synthetic Image Anomalies for Analysis

David Breuss[†1,2] [0009−0007−7743−0243] david.breuss@tuwien.ac.at,
Karel Rusý[†1] [0009−0005−2800−3830] karel.rusy@tuwien.ac.at,
Maximilian Götzinger[3] [0000−0002−1112−141X] maximilian.goetzinger@se.com,
and Axel Jantsch[1,2] [0000−0003−2251−0004] axel.jantsch@tuwien.ac.at

[1] Vienna University of Technology, ICT, Gußhausstraße 27, 1040, Vienna, Austria
[2] Christian Doppler Laboratory for Embedded Machine Learning
[3] Schneider Electric Power Drives GmbH, Ruthnergasse 1, 1210, Vienna, Austria

**Abstract.** Many powerful anomaly detection algorithms are based on machine learning and rely on datasets for training and evaluation. However, anomalous samples are often rare in real-world datasets and might not be representative of anomalies encountered in the field. In this paper, we propose a synthetic anomaly generation methodology that focuses on generating large numbers of synthetic anomalies in images with defined variances in size, shape, and texture, achieving higher diversity scores than the state-of-the-art. To demonstrate the value of the proposed generation methodology for in-depth performance analysis, we generate anomalies in three *MVTec AD* datasets, which we then use to analyze and evaluate several anomaly detection algorithms. While all analyzed anomaly detection algorithms showed strong recall rates on these datasets, significant sensitivity differences regarding an anomaly's size, shape, and texture are observable through the analysis with our synthetic datasets. While we observed some algorithms' robustness towards different anomaly shapes and textures, others showed differences in recall rates of up to 80 percentage points for some pixel manipulation methods. The results demonstrate the value of our synthetic anomalies, as they boost the capability to scrutinize anomaly detection algorithms.

**Keywords:** Synthetic Data, Anomalies, Image Generation, Analysis

## 1 Introduction

An efficient and high output with less manual effort drives today's economy. Monitoring images of goods, systems, and critical infrastructure is often crucial for detecting anomalies and ensuring quality and operability [23]. Increasing computational power in recent decades has enabled the development of new anomaly detection algorithms [24,2], many of which rely on deep learning and datasets used for training [31]. These datasets may require anomaly-free data, anomalous

[†] Both authors contributed equally to this work.

data only, or a mixture [5,16]. Regardless, the quality and quantity of data significantly impact algorithm performance [1]. Even for algorithms not trained with a dataset, anomalous data is needed to measure performance metrics like recall rate or accuracy. However, due to their rarity, annotation effort, and associated costs, gathering labeled anomaly data is often difficult in real-world datasets [18]. Additionally, obtaining high-quality, real-world data is challenging due to privacy concerns, data scarcity, and collection costs. Moreover, even with high-quality data, there is no guarantee that it contains all possible types of anomalies. This scarcity of labeled anomalies emphasizes the need for approaches, such as synthetic anomaly generation, to thoroughly assess the performance of anomaly detection algorithms. Hence, it is essential to develop techniques that enable the controlled introduction of anomalies into the analysis process to ensure the effectiveness of anomaly detection algorithms in real-world scenarios and to avoid leaving crucial blindspots undetected. Several research groups have introduced various anomaly generation methodologies [15,27,12,26,25]. While a few algorithms focus on inserting annotated anomalies from similar datasets into the one of interest [25], others aim to learn the properties and characteristics of already-known anomalies to generate synthetic ones. Although these methods help increase the number of anomalies with specific properties, they rely on some knowledge about anomalies within a dataset. Other methods like *CutPaste* [15] and *NSA* [27] aim to create synthetic anomalies from anomaly-free images for self-supervised training of models. Because the generated anomalies come from patches of anomaly-free data, these methods do not introduce a large variety of textures and shapes. This paper proposes Synthetic Anomaly Generator (SYNAGEN)[1], which enables researchers to enrich existing image datasets with synthetic anomalies with vast variances in size, shape, and texture to reduce the risk of biases towards certain features. We focus on generating surface anomalies for three industrial texture datasets to demonstrate SYNAGEN's value for analysis. In order to ensure the generation of different-looking anomalies while still having control over the anomalies' sizes, shapes, and textures, our SYNAGEN approach is based on three steps, namely:

- a randomized mask generation based on four different complex shapes, modified by noise and added artifacts,
- five different kinds of pixel manipulation modes, and
- a smoothing operation to let the anomaly appear with realistic gradients.

## 2 Related Work

Natural anomalies are often challenging to obtain [32]. Traditional data augmentation methods like scaling, rotating, and shifting can help address imbalanced datasets, but they often leave known anomaly features unchanged [14,19]. Synthetic anomalies offer a controlled means to create labeled data [28]. One popular technique, the Synthetic Minority Oversampling Technique (SMOTE),

[1] SYNAGEN's code: https://github.com/embedded-machine-learning/synagen

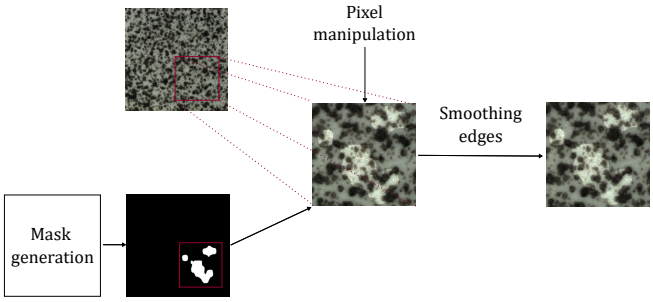



Fig. 1: Overview of SYNAGEN's processing steps.

generates synthetic outliers by interpolating between minority class instances and their nearest neighbors [6]. In [20], Mohammed *et al.* show its effectiveness in improving classification performance. Several variants of SMOTE, such as Borderline-SMOTE and Adaptive Synthetic (ADASYN), have been developed to address specific challenges and data distributions [10,11]. In 2014, Goodfellow *et al.* introduced Generative Adversarial Networks (GANs), which have become a widely used tool in various applications, including image synthesis, super-resolution, and more [17]. For some application domains like fabric defect detection, the consistency in visual defect appearance across fabrics allows the transfer of features and knowledge of one fabric to others [25]. Rippel *et al.* used a GAN-based approach to replicate the anomalies in one type of fabric in others [25]. Since this approach relies on available anomalies and replicates those, lacking representation of previously unseen anomalies is a problem. Salem *et al.* proposed a Cycle-GAN-based method to generate anomalous image data from anomaly-free images [26], yet it still depends on known anomalies. Recently, self-supervised anomaly detection methods, where the training of models only relies on label-free data without any external annotations, have gained more attention in the community [12,15]. Schlüter *et al.* introduced a self-supervision task called Natural Synthetic Anomalies [27] for anomaly detection and localization, which generates anomalies by inserting resized patches from anomaly-free images into random locations. However, since only patches of the original anomaly-free image data are used to generate synthetic anomalies, there is limited variety in anomalous textures and shapes.

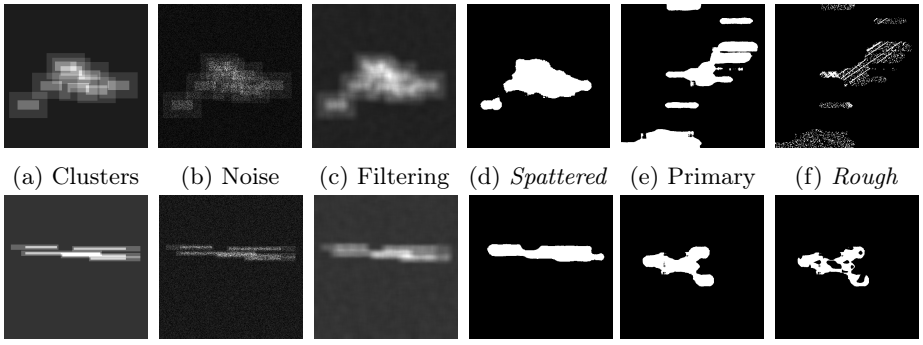
## 3 Synthetic Anomaly Generation

SYNAGEN is a synthetic image anomaly generation tool that modifies regions in anomaly-free input images to create surface anomalies. Fig. 1 illustrates the methodology of SYNAGEN's process, which consists mainly of three steps, as explained below.

### 3.1 Anomaly Mask Generation

Since real-world anomalies appear in various, often unpredictable, forms, creating numerous shapes of different sizes and characteristics is essential. SYNAGEN offers four groups of anomaly masks (*spattered*, *elongated*, *rough*, and *complex*), all based on the same primary mask generation step. While this step fully defines spattered and elongated masks, rough and complex masks rely on an additional mask modification step.

**Primary Mask Generation** Fig. 2 shows example arrangements for a *spattered* mask (top row) and an *elongated* mask (bottom row). The generation process depends on several parameters, such as the maximum possible size, the number of cluster centers, their sizes, and the distance between them. Randomly chosen values for each parameter from predefined uniform distributions ensure the creation of unique shapes. The maximum possible size defines a square of interest (red square in Fig. 1) where SYNAGEN locates the anomaly mask. Cluster centers are rectangles with pixel values of three that are positioned in the center of larger rectangles with pixel values of two. SYNAGEN places these clusters within the square of interest (Fig. 2a and 2g). This square is then multiplied pixel-wise with a square of the same size whose pixel values are sampled from a uniform distribution over $[0, 1)$ (Fig. 2b and 2h). After blurring the resulting product with a Gaussian filter (Fig. 2c and 2i), pixels larger than a threshold value of 1.5 are set to one and all others to zero (Fig. 2d and 2j). This primary mask generation process completely defines the *spattered* and *elongated* masks. The only difference between these two mask groups is the difference in the cluster center rectangle's height-to-width ratio and constraints regarding positioning cluster centers next to each other.



(a) Clusters (b) Noise (c) Filtering (d) *Spattered* (e) Primary (f) *Rough*

(g) Clusters (h) Noise (i) Filtering (j) *Elongated* (k) Primary (l) *Complex*

Fig. 2: The steps for generating a *spattered* (a to d), *elongated* (g to j), *rough* (e and f), and *complex* (k and l) mask.

**Mask Modification** Fig. 2e and 2f show example arrangements for a *rough* mask, and Fig. 2k and 2l for a *complex* mask. For *rough* masks, SYNAGEN multiplies the primary mask (Fig. 2e) with salt noise of varying density and adds straight lines to model cracks and scratches (Fig. 2f). For *complex* masks, a small square filled with pixel values sampled from a uniform distribution over $[0, 1)$ is upscaled to the size of the primary mask's square of interest. After multiplying the primary mask (Fig. 2k) with this up-scaled noise image, SYNAGEN blurs the result with a Gaussian filter (Fig. 2l).

### 3.2 Pixel Manipulation

SYNAGEN provides five different pixel manipulation modes to modify pixels within the anomaly mask:

1. The *transparent* mode multiplies all pixels of each channel of an image by a random factor. This allows the pixels within the anomaly mask to appear darker or brighter than the original and change colors through different factors for each channel. Since all pixels within a channel are changed by the same factor, features and the texture of the original image are still visible after this manipulation (see the first two *wood* anomalies in Figure 4).
2. The first step of the *transparent upscaled noise* mode does the same manipulation as the *transparent* mode, but all color channels are multiplied by the same factor. In a second step, the pixels within the anomaly mask are multiplied pixel-wise with an up-scaled noise image, where the noise pixel values are sampled from a uniform distribution over $[0.5, 1.5)$. The original size of the noise image varies to introduce additional texture differences through the up-scaling. After this pixel manipulation, some of the original texture is still visible but is overlaid with noise (see the first *leather* anomaly in Figure 4).
3. The *transparent color upscaled noise* mode performs the same pixel manipulation as the *transparent upscaled noise* mode except that the factors by which each pixel is multiplied in the first step can be different for each color channel (see the first *tile* anomaly in Figure 4).
4. In the first step of the *mean upscaled noise* mode, all pixel values within the anomaly mask are replaced by their mean value multiplied by a random factor sampled from a uniform distribution. Then, like in the other modes with upsampled noise, all anomaly pixels are multiplied by the upscaled noise image (see the second *leather* anomaly in Figure 4).
5. The *gray upscaled noise* mode replaces all pixels within the anomaly mask with gray upscaled noise. Therefore, no texture of the original image is left within the anomalous region after the *mean upscaled noise* and *gray upscaled noise* manipulation modes (see the second *tile* anomaly in Figure 4).

### 3.3 Smoothing Edges

As the final step, smoothing with a Gaussian filter avoids high gradients at the border between a synthetic anomaly and the original image. The filter alters only pixels of an anomaly's edge and its direct neighbors to preserve the intended change in texture. All other anomaly pixels stay unaltered.

## 4 Datasets for Analysis

*MVTec AD* is a collection of fifteen datasets for benchmarking anomaly detection algorithms focusing mainly on industrial inspection [4]. All these datasets contain anomaly-free training data and separate test data, including anomalous samples. Thus, they are usable for training and evaluating unsupervised algorithms. Each dataset contains between 30 and 141 test images with anomalies and 12 to 58 test images without any anomalies. Because researchers worldwide have used this dataset collection to compare various anomaly detection methods and demonstrate the performance of novel approaches [3,9,27], we also use them to demonstrate SYNAGEN's capabilities and usefulness for an in-depth analysis of five anomaly detection algorithms (see Section 6). For many *MVTec AD* datasets, including all object datasets and the texture datasets *grid* and *carpet*, segmentation masks are necessary to distinguish the objects of interest from the background. Since such masks are not available, and for the benefit of a more detailed analysis, we focus here on three of *MVTec AD*'s texture datasets, namely *wood*, *tile*, and *leather*. We only used SYNAGEN to modify images that do not contain real anomalies from the respective *MVTec AD* dataset's *test* folder. Since all algorithms are exclusively trained with images from the respective *train* folders, none of the analyzed algorithms has seen any of SYNAGEN's input images during training. We created three unique anomalies for each input image with SYNAGEN for every combination of the four shapes and the five different pixel manipulation methods (Section 3). Before any pixel manipulations,

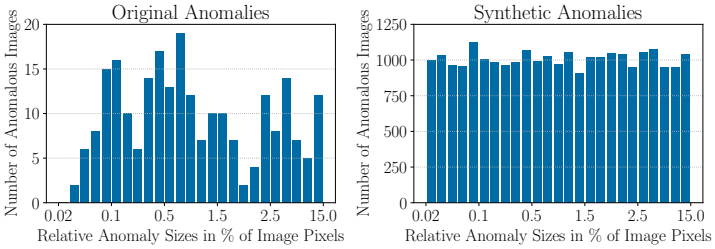



Fig. 3: The combined anomaly size distributions of the original (left) and synthetic extension (right) of *MVTec AD*'s *wood*, *tile*, and *leather* datasets.

SYNAGEN randomly rotates and positions the anomaly masks by drawing an angle and position from uniform distributions over the valid ranges. Anomalies are then down- or upscaled to fit into one of five predefined size ranges, ensuring that they match the user's specified size distribution. The exact size within each range is sampled from a uniform distribution. For all experiments conducted, we defined these ranges as $[0.02\%, 0.1\%)$, $[0.1\%, 0.5\%)$, $[0.5\%, 1.5\%)$, $[1.5\%, 2.5\%)$, and $[2.5\%, 15.0\%]$, indicating the percentage of image pixels altered. Anomaly sizes are evenly distributed between these five bins and sampled from uniform

distributions within the bins. Fig. 3 shows the size distribution of the generated anomalies on the left side. The slight deviations from a uniform distribution result from the last Gaussian filter step. Through this almost uniform distribution, anomalies of different sizes are represented equally in contrast to the size distribution of the original anomalies within the *MVTec AD* dataset. By creating three unique anomalies for every combination of the proposed anomaly masks and pixel manipulation methods and resizing it for the five size ranges, SYNAGEN generated 300 anomalies for every test image without natural anomalies. We did this for the *wood*, *tile*, and *leather* datasets, resulting in 5,700, 9,900, and 9,600 anomalous images, respectively.

## 5 Comparison with State of the Art

Generative approaches that learn from already known real anomalies [25,8,13] aim at replicating these anomalies and, therefore, introduce a bias toward this training data. This bias renders these approaches useless for our analysis because we want to generate anomalies not represented by available anomalous data. Therefore, we compared SYNAGEN's generated synthetic extensions of the *wood*, *tile*, and *leather* datasets within *MVTec AD* (Section 4) to the datasets created by the state-of-the-art approaches CutPaste [15], NSA [27], FPI [29], and DRAEM [33], which like SYNAGEN do not rely on available real anomalous data. For a fair comparison between all approaches, we used all five methods to generate 300 anomalous images from every good image within the test folder of the *MVTec AD wood*, *tile*, and *leather* datasets following a unified size distribution shown in Figure 3.

We quantify the diversity of the synthetic anomalous image dataset extensions with the Learned Perceptual Image Patch Similarity (LPIPS) metric. Zhang et al. introduced LPIPS as a metric for assessing the perceptual similarity between two images [34]. We used *AlexNet* as the feature extractor. For each image within a dataset, we define a random pair to calculate the LPIPS score and interpret the average of calculated LPIPS scores within a dataset as its diversity score. We obtained the $LPIPS_{wood}$, $LPIPS_{tile}$, and $LPIPS_{leather}$ scores by calculating the LPIPS diversity score on the synthetic anomalous extensions of the *MVTec AD wood*, *tile*, and *leather* datasets, respectively. We calculate $LPIPS_{AbsDiff}$ on the dataset, which consists of absolute differences between the synthetic images and their original unaltered versions. Therefore, only the feature differences due to the inserted anomalies contribute to the LPIPS score. LPIPS is based on quantifying the differences in extracted features within two images. Most image pixels stay unaltered through CutPaste, NSA, FPI, DRAEM, and SYNAGEN. Therefore, we can calculate LPIPS exclusively on resized square patches that fully enclose all non-zero pixels within the absolute difference between the synthetic images and their original unaltered versions. These square patches were resized to $256 \times 256$ pixels and used as input images to calculate the $LPIPS^{\square}_{AbsDiff}$ score. Analogous to the $LPIPS^{\square}_{AbsDiff}$ score, the $LPIPS^{\square}_{wood}$, $LPIPS^{\square}_{tile}$, and $LPIPS^{\square}_{leather}$ scores calculate the LPIPS score on the resized

square patches that fully enclose the anomaly within images. Since the $LPIPS^{\square}$ scores aim to eliminate anomaly sizes' influence on diversity scores through focusing and rescaling, they can be interpreted as relative scores regarding anomaly size. The higher the LPIPS scores, the more diverse the according dataset is. Table 1 lists all calculated LPIPS scores for *MVTec AD*'s real anomalies and for

Table 1: LPIPS score listing of *MVTec AD*'s real anomalies and the approaches' generated anomalies.

| Score | MVTec AD | CutPaste | FPI | NSA | DRAEM | SYNAGEN |
|---|---|---|---|---|---|---|
| $LPIPS_{wood}$ | - | 0.291 | 0.286 | 0.296 | 0.321 | **0.343** |
| $LPIPS_{tile}$ | - | 0.358 | 0.358 | 0.376 | 0.372 | **0.397** |
| $LPIPS_{leather}$ | - | 0.227 | 0.228 | 0.239 | 0.249 | **0.257** |
| $LPIPS^{\square}_{wood}$ | 0.403 | 0.448 | 0.406 | 0.379 | 0.453 | **0.647** |
| $LPIPS^{\square}_{tile}$ | 0.490 | 0.571 | 0.579 | 0.592 | 0.542 | **0.663** |
| $LPIPS^{\square}_{leather}$ | 0.514 | 0.486 | 0.475 | 0.458 | 0.478 | **0.612** |
| $LPIPS_{AbsDiff}$ | - | 0.071 | 0.056 | 0.053 | 0.080 | **0.134** |
| $LPIPS^{\square}_{AbsDiff}$ | - | 0.433 | 0.306 | 0.224 | 0.515 | **0.569** |

the five synthetic anomaly generation approaches. For *MVTec AD*'s real anomalies, the $LPIPS_{wood}$, $LPIPS_{tile}$, and $LPIPS_{leather}$ scores are not listed because the deviating anomaly size distributions render these scores meaningless for a fair comparison. However, since the $LPIPS^{\square}$ scores only consider the rescaled regions of images containing anomalies, the influence of an anomaly's size on the $LPIPS^{\square}$ score is minimized. Furthermore, since the absolute difference images between images with and without anomalies only exist for the synthetic images, the $LPIPS_{AbsDiff}$ scores are not applicable to the real *MVTec AD* anomalies.

SYNAGEN achieves the highest diversity scores for all analyzed $LPIPS$ metrics compared to the four state-of-the-art approaches. In the CutPaste, FPI, and NSA approaches, the anomaly shapes are rectangular patches of varying height-to-width ratios, which clearly limits shape diversity. Furthermore, these three approaches replace the pixels within the anomaly shape with other original pixels within the images and do not introduce a large variety of texture changes. Therefore, their diversity scores are consistently lower than SYNAGEN's scores. Although DRAEM introduces various textures from the *Describable Texture Dataset* [7], SYNAGEN still consistently achieves higher diversity scores. One reason for this might be the higher variance in SYNAGENs' shapes since DRAEM uses Perlin noise for every anomaly mask generation. The LPIPS scores suggest that SYNAGEN succeeds at generating more diverse anomalies than the state-of-the-art. Furthermore, *MVTec AD*'s real anomalies achieve significantly lower diversity scores than SYNAGEN's generated datasets. Fig. 4 shows a few example anomalies generated by each analyzed approach.

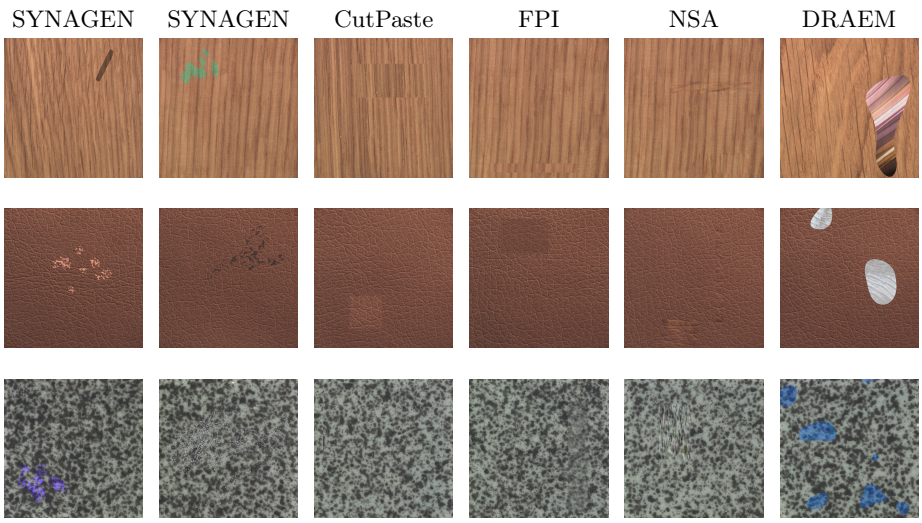



Fig. 4: These figures show examples of generated anomalies by our approach SYNAGEN and state-of-the-art approaches CutPaste, FPI, NSA, and DRAEM.

## 6 Analysis of Anomaly Detection Algorithms

Different metrics are available to analyze anomaly detection algorithms. *Auroc* is often used to compare algorithms on imbalanced datasets since it considers both the recall and false positive rates. *Auroc* is defined as the area under the recall rate versus the false positive rate curve. It depends on anomaly-free images in the dataset and is influenced by the ratio between anomalous and anomaly-free images. Therefore, our analysis focuses on recall rate as a metric while setting a fixed false positive rate for the experiments on a given dataset. To demonstrate the benefit of synthetic anomalies created by SYNAGEN, we evaluated five anomaly detection algorithms—*EfficientAD* [3], *MSFlow* [35], *ReContrast* [9], *DDAD* [21], and *RD++* [30]—on three subsets of the *MVTec AD* dataset [4], namely *wood*, *tile*, and *leather* datasets. We trained the models with the *train* directories through unsupervised training without any anomalous images of the respective *MVTec AD* dataset. On *MVTec AD*, these algorithms achieve an *auroc* for anomaly detection of 99.8%, 99.7%, 99.5%, 99.5%, and 99.44% when evaluated on all images within the *test* folders of the *MVTec AD* datasets [22]. We chose to compare these five algorithms' performances with each other because they are based on different detection approaches and achieve similar *auroc* scores on *MVTec AD*. Although the algorithms' anomaly localization capabilities could also be analyzed using this methodology, we exclusively focused on anomaly detection for a more detailed analysis. Through systemic variation of anomaly characteristics, our synthetic anomaly creation methodology enables the analysis of a detection algorithm's sensitivity. The large variety

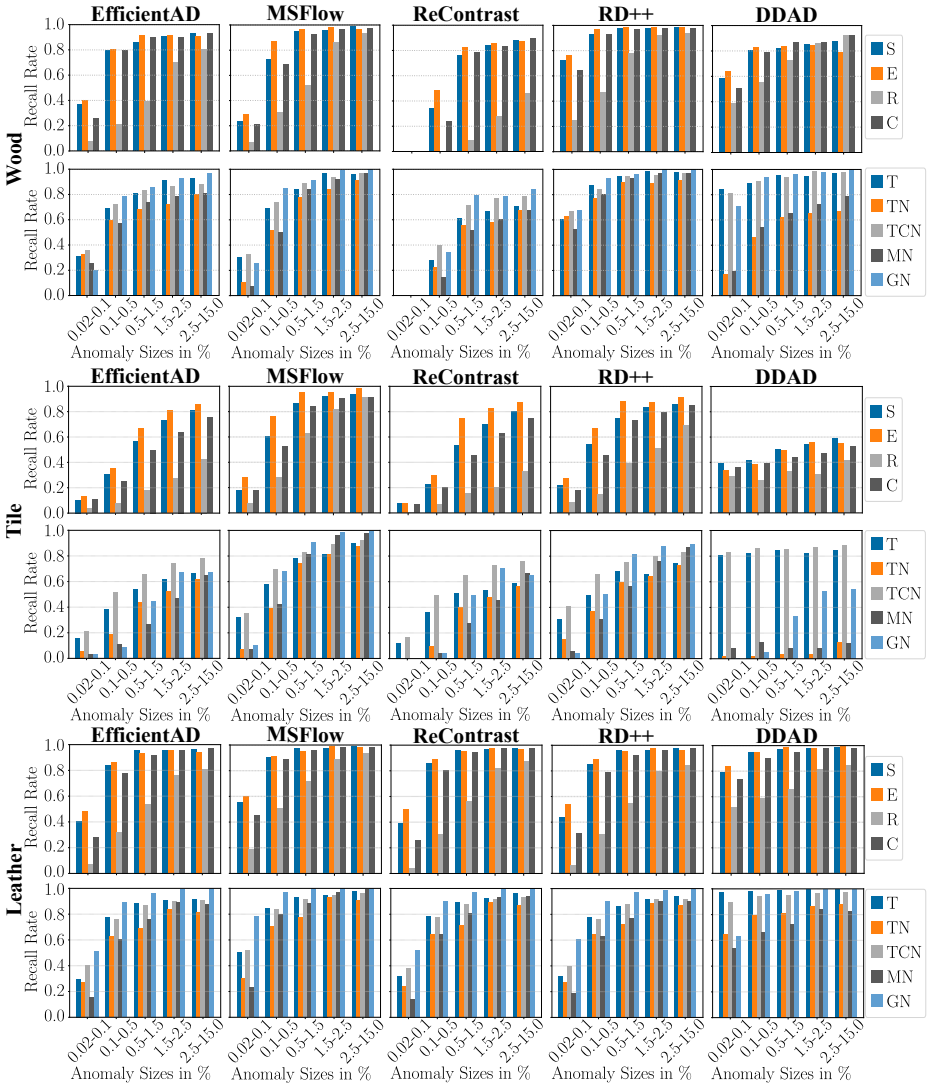



Fig. 5: Algorithms' recall rates for anomalies of different sizes, shapes, and pixel manipulation methods in the *wood*, *tile*, and *leather* datasets. T, TN, TCN, MN, and GN refer to *transparent*, *transparent upscaled noise*, *transparent color upscaled noise*, *mean upscaled noise*, and *gray upscaled noise*, and S, E, R, and C to *spattered*, *elongated*, *rough*, and *complex*, respectively.

in anomaly sizes, shapes, and textures helps to minimize the risk of analysis bias towards specific anomaly features. Using recall rate as a comparison metric allows us to analyze an algorithm's performance on subsets of the datasets without focusing on anomaly-free test data. To ensure a fair comparison, we set a uniform threshold for each algorithm to maintain consistent false positive rates; i.e., one of the corresponding original dataset's anomaly-free test images is falsely classified. Figure 5 demonstrates each algorithm's sensitivity to anomalies' size, shape, and texture within SYNAGEN's *wood*, *tile*, and *leather* datasets. The first two rows, the middle two rows, and the last two rows show the recall rates measured on the *wood*, *tile*, and *leather* datasets, respectively. The respective first rows for each dataset present the recall rates for different shapes. Two facts apply to all algorithms: the anomaly size always positively correlates with the recall rate, and anomalies with a *rough* shape lead to the lowest recall rates. The most significant difference between the *rough* and other shapes is that *rough* shapes consist of many small anomalies since masks of this type consist of thin lines and small spots (i.e., Fig. 2f). *EfficientAD* and *ReContrast* show the largest differences in recall rate between the *rough* and other shapes. Since such *rough* shapes consist of multiple small anomalous regions, these results are consistent with *EfficientAD*'s and *ReContrast*'s low recall rates for smaller anomalies. The second, fourth, and sixth rows in Figure 5 show the recall rates of all analyzed algorithms for the five proposed pixel manipulation methods and different anomaly sizes on the respective dataset. Interestingly, an anomaly's texture has a larger influence on the recall rate of smaller anomalies.

Despite the generally positive correlation between anomaly size and recall rate, this correlation is weaker for *DDAD*'s detection of anomalies with *transparent* and *transparent color upscaled noise* pixel manipulations. Moreover, *DDAD* is significantly more sensitive to the *transparent*, *transparent color upscaled noise* and *gray upscaled noise* pixel manipulations. Therefore, *DDAD*'s recall rates for these types of pixel manipulations are high, even for the smallest anomalies. The *transparent upscaled noise* and *mean upscaled noise* pixel manipulations do not introduce significant relative differences between the color channels like the other three techniques. Therefore, these experiments suggest that *DDAD* is more sensitive to such color changes and struggles when only the brightness is altered. Figure 6 plots the recall rates of all analyzed algorithms on the three generated anomalous datasets for five different false positive rates. For each dataset, we considered all synthetic anomalies without distinction between shapes, textures, or sizes to calculate these recall rates. The false positive rates differ for the three datasets because we defined the threshold for each algorithm and dataset to allow for exactly one, two, three, four, or five false positives. Because each of *MVTec AD*'s datasets has a different number of anomaly-free test images, the corresponding false positive values vary. Since higher false positive rates correspond with lower threshold values of the algorithms, a positive correlation exists between the recall rates and the false positive rate. Although all algorithms achieve similar *auroc* scores between 99.44% and 99.8% on the original *MVTec AD* dataset, resulting from almost identical recall rates, there are sig-

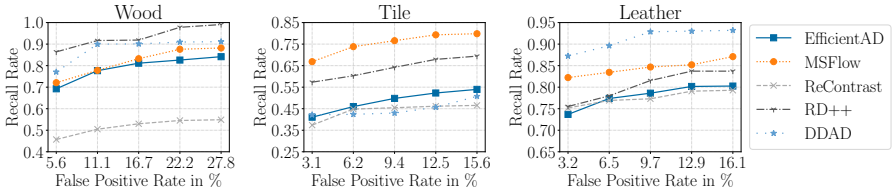



Fig. 6: Recall rates of the analyzed algorithms on our synthetic anomalous *wood*, *tile*, and *leather* datasets. The results show significant performance differences.

nificant differences in the overall recall rates between the five algorithms on our synthetic datasets. The algorithms' sensitivities towards size, shape, and texture mostly show the same trends on these datasets. However, for our synthetic *tile* dataset, we observed a difference in *DDAD*'s recall rate between the anomalies with different pixel manipulation methods of up to 80 percentage points for all analyzed ranges of anomaly sizes. While *MSFlow* achieves similar results on all three datasets, the other algorithms score significantly lower recall rates for the *tile* datasets. The reason for both of these statements might be that the tile images consist of multiple gray pixel regions of different brightness values and sizes. All algorithms struggle most with small *transparent upscaled noise*, *mean upscaled noise*, and *grey upscaled noise* textures within the *tile* dataset. These pixel manipulations lead to changes in brightness rather than color, which are harder to detect. *DDAD*'s recall rates for the *transparent* and *mean upscaled noise* textures within the *tile* dataset even stay below 13% for all size ranges.

Compared to the other algorithms, *DDAD* shows a higher overall recall rate on the *leather* dataset but struggles with detecting our synthetic *tile* anomalies. One reason for this might be the lower perceptual diversity in the leather images compared to the tile images, which is supported by $LPIPS_{leather} < LPIPS_{tile}$ (see Table 1). *DDAD* might profit from such datasets with lower perceptual diversity since one of its mechanisms tries to capture the perceptual similarity of extracted features as a metric for anomalies [21]. Our experiments show that *ReContrast*'s overall recall rate on our synthetic *wood* dataset is more than 20 percentage points lower than the recall rates of other algorithms. *ReContrast*'s recall rate of smaller anomalies, especially on the *wood* and *tile* datasets, is clearly outperformed by the other algorithms. Since such small anomalies are underrepresented in the original *MVTec AD* datasets, this lack of sensitivity stays hidden when only evaluated with the labeled real anomalies. We can obtain more detailed information regarding each algorithm's recall rate for specific anomaly types by simultaneously fixing the shape and pixel manipulation method and focusing on different size ranges. The left graph in Figure 7 demonstrates this on the *leather* dataset with the shape *rough* and pixel manipulation method *mean upscaled noise*. On this specific type of anomaly, *MSFlow* outperforms the other algorithms for relative anomaly sizes larger than 0.1%. *MSFlow*'s asymmetrical parallel flows and the fusion flow for multi-scale perception exchange target the

anomaly size variation problem [35]. Since the *rough* shapes typically include multiple small-scale anomalies, this approach enables *MSFlow* to outperform the other algorithms. Besides the shape, pixel manipulation method, and size of an anomaly, the mean background brightness and mean relative brightness difference due to an anomaly can also affect the recall rate. The right graph of Figure 7 shows how the relative brightness difference resulting from anomalies with a relative size between 0.02 and 0.1%, an *elongated* shape, and the *transparent* pixel manipulation method affects the algorithm's recall rate. The relative brightness difference significantly impacts the recall rates, and *DDAD* outperforms the other algorithms for this specific type of anomaly in the leather dataset. Besides the pixel differences between the input and reconstructed image, which partly rely on perceptual color variations, *DDAD* also considers features extracted by deep neural networks to capture perceptual similarity [21]. This perceptual similarity comparison seems to enable *DDAD* to detect this type of anomaly even with small relative brightness differences between 10 and 20%.

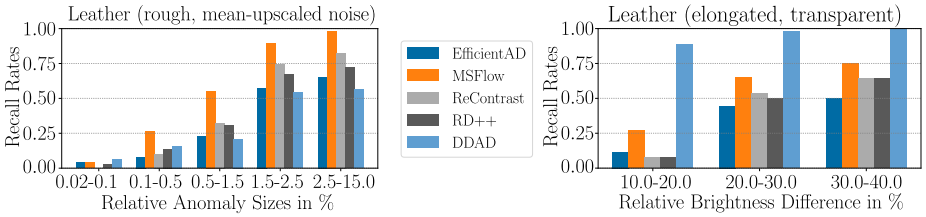



Fig. 7: These graphs show how the relative anomaly size and brightness difference of specific anomaly types influence the algorithm's recall rates on the *leather* dataset. In the right graph the relative anomaly size is fixed to 0.02-0.01%.

## 7 Conclusion and Outlook

The combination of increasing computational power and recent progress in machine learning and computer vision led to novel anomaly detection approaches that enable the automatic inspection of goods, systems, and critical infrastructure. When annotated anomalies are scarce, synthetic anomalous data can be used to enrich datasets for more detailed algorithm analysis. Our proposed Synthetic Anomaly Generator (SYNAGEN) can generate anomalies with a wide range of shapes, sizes, and textures to analyze algorithms. Combining a randomized shape generation process and several pixel manipulation techniques for random textures enables an in-depth analysis of anomaly detection approaches while reducing the risk of hidden biases in analysis. With SYNAGEN, we generated thousands of surface anomalies with a predefined size distribution for *MVTec AD*'s *wood*, *tile*, and *leather* datasets. SYNAGEN achieved higher diversity scores than state-of-the-art approaches on all analyzed datasets. We analyzed five anomaly detection algorithms based on different approaches on these

three synthetic datasets to highlight how SYNAGEN's synthetic anomalies enable a detailed analysis of the algorithm's sensitivities regarding the size, shape, and texture of anomalies. By comparing the performance of these five detection algorithms on our synthetic data, we observed significant differences in recall rates for different shapes, textures, and sizes. Experiments showed differences in recall rates of some anomaly textures of up to 80 percentage points. Similarly, some algorithms' recall rates change by up to 70 percentage points for certain anomaly shapes of the same size range. When the analysis only considers the original *MVTec AD* test datasets, these significant performance differences stay undetected due to the low variability and number of anomalies.

Even though we solely demonstrated SYNAGEN's capabilities for analyzing and evaluating anomaly detection algorithms, self-supervised anomaly training methodologies, where synthetic anomaly images are utilized during training, might also benefit from SYNAGEN's anomalies. Although we demonstrated SYNAGEN for three texture datasets, this tool can also be used in more complex scenarios by exclusively inserting anomalies in optionally defined regions by an input mask and leaving the rest of the image (e.g., the background) unaltered. Adding additional shape generation and pixel manipulation modes to SYNAGEN could further increase anomaly diversity. Furthermore, this could enable SYNAGEN to generate anomalies with characteristics specialized for other use cases and fields other than surface anomaly detection.

**Acknowledgments.** This study was funded by the Austrian Research Promotion Agency (FFG) under contract 880842 and by the Christian Doppler Research Association (CDG) as part of the project Embedded Machine Learning.

## References

1. Abufadda, M., et al.: A survey of synthetic data generation for machine learning. In: ACIT (2021)
2. Ayub Khan, A., et al.: Machine learning in computer vision: A review. ICST Transactions on Scalable Information Systems **8** (04 2021)
3. Batzner, K., et al.: Efficientad: Accurate visual anomaly detection at millisecond-level latencies (2023)
4. Bergmann, P., et al.: Mvtec ad — a comprehensive real-world dataset for unsupervised anomaly detection. In: CVPR (2019)
5. Chandola, V., et al.: Anomaly detection: A survey. ACM Comput. Surv. **41**(3) (jul 2009)
6. Chawla, N.V., et al.: SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research **16**, 321–357 (Jun 2002)
7. Cimpoi, M., et al.: Describing textures in the wild. In: CVPR (2014)
8. Duan, Y., et al.: Few-shot defect image generation via defect-aware feature manipulation (2023)
9. Guo, J., et al.: Recontrast: Domain-specific anomaly detection via contrastive reconstruction (2023)
10. Han, H., et al.: Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In: Advances in Intelligent Computing. Lecture Notes in Computer Science (2005)

11. He, H., et al.: ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence) (2008)
12. Hojjati, H., et al.: Self-supervised anomaly detection: A survey and outlook (2023)
13. Hu, T., et al.: Anomalydiffusion: Few-shot anomaly image generation with diffusion model (2023)
14. Kwasigroch, A., et al.: Deep convolutional neural networks as a decision support tool in medical problems – malignant melanoma case study. In: Trends in Advanced Intelligent Control, Optimization and Automation (2017)
15. Li, C.L., et al.: Cutpaste: Self-supervised learning for anomaly detection and localization. In: CVPR (2021)
16. Li, T., et al.: Deep unsupervised anomaly detection. In: WACV (2021)
17. Liu, Y., et al.: Generative Adversarial Active Learning for Unsupervised Outlier Detection (Mar 2019), http://arxiv.org/abs/1809.10816
18. Lu, Y., et al.: Machine Learning for Synthetic Data Generation: A Review (2023)
19. Mikołajczyk, A., et al.: Data augmentation for improving deep learning in image classification problem. In: IIPhDW (2018)
20. Mohammed, A.J.: Improving Classification Performance for a Novel Imbalanced Medical Dataset using SMOTE Method. International Journal of Advanced Trends in Computer Science and Engineering (Jun 2020)
21. Mousakhan, A., et al.: Anomaly detection with conditioned denoising diffusion models (2023)
22. MVTec AD, p.: Mvtec ad benchmark on paperswithcode.com. https://paperswithcode.com/sota/anomaly-detection-on-mvtec-ad (2023), accessed: 2023-10-25
23. Newman, T.S., et al.: A survey of automated visual inspection. Computer Vision and Image Understanding **61**(2), 231–262 (1995)
24. Patel, K.B.: A review: Machine vision and its applications. IOSR Journal of Electronics and Communication Engineering **7**, 72–77 (2013)
25. Rippel, O., Müller, M., Merhof, D.: Gan-based defect synthesis for anomaly detection in fabrics. In: ETFA (2020)
26. Salem, M., et al.: Anomaly generation using generative adversarial networks in host based intrusion detection. CoRR **abs/1812.04697** (2018)
27. Schlüter, H.M., et al.: Natural synthetic anomalies for self-supervised anomaly detection and localization. In: ECCV (2022)
28. Steinbuss, G., et al.: Benchmarking unsupervised outlier detection with realistic synthetic data. ACM Trans. Knowl. Discov. Data **15**(4) (apr 2021)
29. Tan, J., et al.: Detecting outliers with poisson image interpolation. In: MICCAI (2021)
30. Tien, T.D., et al.: Revisiting reverse distillation for anomaly detection. In: CVPR (2023)
31. Wankhede, S.B.: Anomaly detection using machine learning techniques. In: I2CT (2019)
32. Yang, J., et al.: Visual anomaly detection for images: A systematic survey. Procedia Computer Science **199**, 471–478 (2022)
33. Zavrtanik, V., et al.: DrÆm – a discriminatively trained reconstruction embedding for surface anomaly detection. In: ICCV (2021)
34. Zhang, R., et al.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)
35. Zhou, Y., et al.: Msflow: Multi-scale flow-based framework for unsupervised anomaly detection (2023)