Hindawi
International Journal of Intelligent Systems
Volume 2023, Article ID 7773481, 18 pages
https://doi.org/10.1155/2023/7773481

Research Article

GRD-Net: Generative-Reconstructive-Discriminative Anomaly Detection with Region of Interest Attention Module

Niccolò Ferrari,[1,2] **Michele Fraccaroli**,[1] **and Evelina Lamma**[1]

[1]*Department of Engineering, University of Ferrara, Via Saragat 1, 44122 Ferrara, Italy*
[2]*Bonfiglioli Engineering, Via Amerigo Vespucci 20, 44124 Ferrara, Italy*

Correspondence should be addressed to Niccolò Ferrari; niccolo.ferrari@unife.it

Received 20 December 2022; Revised 3 August 2023; Accepted 7 August 2023; Published 2 September 2023

Academic Editor: Mohammad R. Khosravi

Anomaly detection is nowadays increasingly used in industrial applications and processes. One of the main fields of the appliance is the visual inspection for surface anomaly detection, which aims to spot regions that deviate from regularity and consequently identify abnormal products. Defect localization is a key task that is usually achieved using a basic comparison between generated image and the original one, implementing some blob analysis or image-editing algorithms in the postprocessing step, which is very biased towards the source dataset, and they are unable to generalize. Furthermore, in industrial applications, the totality of the image is not always interesting but could be one or some regions of interest (ROIs), where only in those areas there are relevant anomalies to be spotted. For these reasons, we propose a new architecture composed by two blocks. The first block is a generative adversarial network (GAN), based on a residual autoencoder (ResAE), to perform reconstruction and denoising processes, while the second block produces image segmentation, spotting defects. This method learns from a dataset composed of good products and generated synthetic defects. The discriminative network is trained using a ROI for each image contained in the training dataset. The network will learn in which area anomalies are relevant. This approach guarantees the reduction of using preprocessing algorithms, formerly developed with blob analysis and image-editing procedures. To test our model, we used challenging MVTec anomaly detection datasets and an industrial large dataset of pharmaceutical BFS strips of vials. This set constitutes a more realistic use case of the aforementioned network.

1. Introduction

Semisupervised computer vision is a task increasingly used in the industrial sector. The reasons are to be found in its flexibility and in its capability to generalize when a new anomaly is seen. Moreover, despite the good performance of supervised approaches in computer vision fields, but requiring a large number of examples during the training phase, the previously mentioned approach requires only a significant number of nominal examples to define their distribution. Regarding defects, on the other hand, it requires just a little set of anomalies, used for testing purposes and in some cases to define an anomaly threshold.

In real cases, on production lines, the availability of regular products is the vast majority compared to anomalies. For this reason, the training dataset would be extremely unbalanced in favor of nominal examples. This makes it difficult performing a good training supervised model. Moreover, it is often required to locate the defect within the image because usually the defective portion covers a small area of the whole surface. To give an example, on pharmaceutical vials, defects are frequent little scratches, very small black spots, or some alien particles deposited on the surface of products, which are usually between 100 and 1000 μm. This target could be more easily reached with semisupervised anomaly detection architecture and more specifically with a reconstruction-based approach, in which the network gives you a reconstruction of the image without the anomalous area.

Reconstructive methods include autoencoder (AE) [1, 2], variational autoencoder (VAE) [3, 4], and generative adversarial network (GAN) [5]. They have been thoroughly

investigated since they make it possible to learn a robust reconstruction subspace using only images without anomalies. Thanks to the incapability to rebuild anomalous regions, which were not contained within nominal images during training, the network fails to reproduce the out-of-distribution area. For this reason, it is possible to detect discrepancies between the two images by thresholding, for example, the absolute value of the difference between them. This is the most immediate and simple method for performing final classification, but it is a nonparametric approach for anomaly localization, and in some cases, discrimination could be erroneous in some noisy cases because the sum of all the small differences could exceed the threshold. In other cases, it could be inaccurate due to the lack of comprehension of differences between the two images, which is left to a simple threshold.

In addition to this, our generative-reconstructive-discriminative anomaly detection with region of interest attention module network (GRD-Net) aims to minimize the development of preprocessing algorithms that are used to locate the portion of the image in which they search for anomalies. In classical reconstruction-based methods, the generation of anomaly map, as mentioned above, is left to a threshold-based classifier, so it is not possible to turn attention to one or more specific ROIs. Embedding similarity-based approach constitutes another big family of architectures, offering encouraging results. However, due to the lack of comprehensibility of the result and learning process, it becomes more difficult to impose an ROI, which draws attention to defects.

For all the aforementioned reasons, a second network, chained to the first generative part, is required to achieve the intended results. This work is heavily inspired by the discriminatively trained reconstruction anomaly embedding model (DRÆM) [6]. DRÆM works by learning a joint representation of an anomalous image and its anomaly-free reconstruction and simultaneously learning a decision boundary between anomalous and positive examples. This method enables direct anomaly localization avoiding the implementation of some postprocessing techniques.

The DRÆM is based on a first reconstructive network (an autoencoder) and a discriminative network. The first network is trained to identify and reconstruct anomalies, maintaining the nonanomalous regions of the input image. The second network combines original and reconstructed appearance to learn joint-anomaly inclusion reconstruction, to produce accurate anomaly segmentation maps [6].

In the context of this work, the autoencoder that defines the reconstructive network is replaced with GANomaly [7]. GANomaly is a generative adversarial network (GAN) [5] architecture that simultaneously learns how to create a high-dimensional image space and infer a latent space. The model can map the input image to a lower dimension vector using encoder-decoder-encoder subnetworks, which are then used to reconstruct the generated output image. This generated image is mapped to its latent representation by the additional encoder network. Learning the data distribution for the normal samples is aided by minimizing the distance between these images and the latent vectors during training. The generative part of the GAN is constituted by a fully convolutional residual autoencoder [8], that, as mentioned by Wickramasinghe et al., residual blocks help to prevent the gradient vanishing on deep convolutional networks, thus avoiding the deterioration of learned embedded-representations.

In this work, we summarize the following:

(1) The generalization capability of the GANomaly architecture [7] with the denoising ability derived from the DRÆM architecture [6] is merged in the reconstructive part of the model

(2) The reconstructive autoencoder is residual and fully convolutional [8], improving the stability of the learning process

(3) An attention module that uses a ROI is added for each example during the training phase to learn the area where to focus the segmentation of the abnormal area in the discriminative part of the model

The first network rebuilds the original image in a better and more precise way with a more performing and stable training phase, regarding the two reference models GANomaly and DRÆM. This is due to the residual autoencoder with the GAN structure and the mask superimposed obtained by adding Perlin noise to the input. This technique challenges the network not only to rebuild the input image as it is but also to regenerate the hidden part by the noise in a coherent way. The second block identifies the area where the defect is located, which is a specification required in most industrial applications, with a ROI-based attention module. Defining a ROI for each training example lets the network learn the important area of the product where to look for defects, using the original and the reconstructed image by the first block. In this way, the second net generalizes and spots the ROI in the new input images during production, excluding the research of defects outside. This is a very important result because often we need to spot defects within a region of interest (ROI), excluding the more chaotic and false-reject-prone areas outside.

The rest of the paper is organized as follows: Section 2 describes related works. Section 3 presents the background knowledge necessary for the correct understanding of this work (Sections 3.1 and 3.2) and our contribution (Section 3.3). Section 4 illustrates the experiments and the results obtained on the various datasets. Finally, in Section 5, we present conclusions and future work.

2. Related Work

Many surface anomaly detection techniques exploit the reconstruction-based approach. This approach is based on image reconstruction and identifies anomalies working on image reconstruction errors [7, 9, 10]. Typically, neural networks such as autoencoders (AEs) [10, 11], variational autoencoders (VAEs) [12], and generative adversarial networks (GANs) [7, 13–15] are used for image reconstruction as described in [10]. The finding of an anomaly is generally based on the quality of image reconstruction.

Reconstruction-based methods can use the structural similarity [10] or the pixel-wise reconstruction error [16] as the anomaly score to localize anomalies. A visual attention map created from the latent space can also be used as the anomaly map [12]. Another reconstruction-based model that implements a segmentation structure based on transformer is RDAD [17]. The reconstruction-based approaches are easily interpretable, but their performance is constrained by the fact that AE can occasionally produce good reconstruction outcomes for anomalous images as well [18]. A good comparison and analysis of the different techniques was described by Xia et al. [15], explaining the benefits of a semisupervised machine learning architecture for reconstruction-based methods, but also comparing some embedding similarity-based methods.

Another important family of methods in the field of anomaly detection is, precisely, the embedding similarity-based approach. These techniques extract useful vectors describing an entire image for anomaly detection [19, 20] or an image patch for anomaly localization [21] using deep neural networks. However, in several works based on embedding similarity-based methods, it offers encouraging results but frequently lacks interpretability. It is impossible to identify the specific aspect of an anomalous image that contributed to its anomaly score. The anomaly score is, in this case, the distance between the embedding vectors of a test image and the reference vectors representing normality from the training dataset. The normal reference can be defined as the center of a sphere that contains embedding from normal images or the entire set of normal embedding as in the case of SPADE [22]. Another interesting approach that works with patch embedding is PaDiM [23]. Normal class in PaDiM is described through a set of Gaussian distributions that utilize the pretrained convolutional neural network (CNN) to model correlations between semantic levels. Heavily related to SPADE and PaDiM, there is PatchCore [24] that uses a memory bank with neighborhood-aware patch-level features in order to increase performance. In addition, corset subsampling of the memory bank ensures low inference cost at higher performance. A further subcategory of methods, however, based on embedding similarity-based approach, is the one based on generative models called normalizing flows (NFLOW) [25]. The main advantage of NFLOW models is the ability to estimate the exact likelihoods for out-of-distribution examples compared to other generative models [26–28]. Notable works in the NFLOW category can be the system developed by Rudolph et al. called DifferNet [29], the work of Gudovskiy et al. called CFLOW-AD [30], and the more recent work of Jaehyeok Bae et al. called PNI [31], which takes into account the position and neighborhood information on the distribution of normal features.

Knowledge distillation techniques are also widely used in anomaly detection tasks, especially when we are dealing with large images, as in the work of Paul et al. [32]. This matter is examined in the work also written by Paul Bergmann et al.[33], in which anomalies are divided into logical and structural. Noteworthy is also the knowledge distillation-based work of Kilian Batzner et al. [34] where processing time plays a central role in the problem definition because more and more often lots of real-time applications use unsupervised machine learning algorithms for anomaly detection tasks.

Reconstruction-based anomaly detection approaches are widely used in different areas with other types of data, such as time series data. In these cases, conventional threshold-based anomaly detection methods are inadequate, as mentioned by Li et al. [35]. To handle this type of data, an LSTM-RNN model must be introduced into the GAN or VAE-GAN architecture [36, 37] with an encoder-decoder-encoder shape. Such data may be derived from industrial processes [38], where it is often difficult to obtain balanced data between regular and abnormal data. Also, it can be obtained by smart grids [39, 40], where it is mandatory to monitor data for security tasks but equally difficult to handle such big data without artificial intelligence algorithms; finally, data could consist of video streams [41].

A special mention should be made to the work of Zavrtanik et al. [6] as this work is largely based and inspired by DRÆM. This work exploits a reconstruction and a discriminative network to segment artificial noise. The output of DRÆM is an anomaly detection mask and the anomaly score. The anomaly mask can be used to estimate the image-level anomaly score. The maximum value of the smoothed anomaly score map is used to calculate the final score.

3. Methods

To explain our approach, we firstly introduce DRÆM and GANomaly as they are the knowledge base necessary for understanding the rest of the paper.

In this section, we summarize the following:

(1) DRÆM architecture [6], which is the starting point of our improvements

(2) GANomaly architecture [7], which extends with GAN's, benefits the DRÆM architecture, with special attention to GAN structure and training loop

(3) The generative-reconstructive-discriminative network (GRD-Net) architecture, with the attention module based on ROIs

3.1. DRÆM. As mentioned before, DRÆM is an anomaly detection framework based on two different subnetworks. The first subnetwork (called reconstructive subnetwork) is trained to recognize anomalies and reconstruct them while keeping the portions of the input image that are not anomalous. The second network learns joint-anomaly inclusion reconstruction to create accurate anomaly segmentation maps by fusing the original and reconstructed appearance.

Instead of generating simulations that accurately reflect the actual appearance of the anomaly in the target domain, DRÆM instead creates just-out-of-distribution appearances that allow learning the proper distance function to identify the anomaly by its departure from normality. This paradigm is used in the proposed anomaly simulator. The images with

artificial anomalies are generated through Perlin noise generator [42] to generate a variety of anomaly shapes (see Figure 1(a)). The generated image is then binarized by a threshold into an anomaly map using uniformly random samples. Then, merging the anomaly map with random RGB pixels, we obtain the final noise (see Figure 1(b)) to be added to the images of the dataset, as can be seen in Figure 1(c). Thus, this process creates training sample triplets with the original image that is free of anomalies, the augmented image that contains simulated anomalies, and the pixel-perfect anomaly mask.

The reconstructive subnetwork of DRÆM performs an image denoising task. It is trained to reconstruct the original image from the artificial corrupted version produced by the process described above. The discriminative subnetwork is a U-Net-like neural network that takes in inputting the channel-wise concatenation of the reconstructive subnet output and the original image. This second subnetwork learns to segment the Perlin noise applied to the original image instead of similarity functions such as SSIM [43].

The output of the discriminative subnetwork is an anomaly detection mask. This mask can be interpreted for the image-level anomaly score estimation. The anomaly mask is smoothed by a convolutional filter. The final anomaly score is computed by taking the maximum value of the smoothed anomaly score map.

3.2. GANomaly

3.2.1. Adversarial Autoencoders. An autoencoder (AE) [44] is a neural network that has been trained to attempt to replicate its input to its output. The two components of this network are an encoder (E) that maps the input into latent space h and a decoder (D) that reconstructs the input from the latent space. The ability to constrain h to be smaller than x and the input copying task are where AE's potential lies (in this case, we talk about undercomplete AE). The network is forced to recognize the most crucial aspects of the input data when learning an undercomplete representation. This procedure can be carried out by minimizing the network's penalty function when it is far from x. To outperform the standard AEs, we can think to train an AE in an adversarial environment [45]. Training AEs with adversarial setting improves reconstruction while also giving the user more control over latent space [46, 47].

3.2.2. Generative Adversarial Networks. GANs are an unsupervised machine learning approach developed for the task of generating synthetic data [5]. Specifically, the first purpose of the GANs was to generate realistic synthetic images. The concept is that during training, two networks—the discriminator and the generator—compete with one another so that the former attempts to generate an image while the latter determines whether it is real or fake. The generator that is similar to a decoder learns the distribution of input data from a latent space.

3.2.3. GANomaly Architecture and Training. The GANomaly architecture contains two encoders, a decoder, forming an encoder-decoder-encoder structure, and discriminator networks [7]. The first encoder-decoder subnetwork in an AE works as the generator of the model. The generator uses an AE network to reconstruct the input image x after learning how to represent the input data. The second encoder of the encoder-decoder-encoder structure is a network that compresses the reconstructed image $\widehat{x}$. This encoder has the same architecture on the previous encoder but with different parametrization. This encoder explicitly learns to minimize the distance with its parametrization. This minimization is used during the test to perform anomaly detection. The discriminator network aims to classify the input x and the output $\widehat{x}$ as real or fake.

GANomaly is trained by minimizing a loss consisting of three components: the adversarial, contextual, and encoder losses. Adversarial loss ($\mathscr{L}_{\text{adv}}$) is calculated for the discriminator, and it is used to reduce the instability of GAN training. Contextual loss ($\mathscr{L}_{\text{con}}$) is used to add the contextual information to the final loss. This subloss consists of the sum of the $\mathscr{L}_1$ distance, between the input image x and the rebuilt image $\widehat{x}$, and the SSIM loss: $\mathscr{L}_{\text{ssim}} = 1 - \text{SSIM}$ score, also calculated between x and $\widehat{x}$. So final $\mathscr{L}_{\text{con}}$ becomes

$$\mathscr{L}_{\text{con}} = \omega_a \mathscr{L}_1 (x, \widehat{x}) + \omega_b \mathscr{L}_{\text{ssim}} (x, \widehat{x}). \tag{1}$$

Finally, the encoder loss ($\mathscr{L}_{\text{enc}}$) is used to minimize the distance between bottleneck features of the input and the encoded features of the generated image. Then, the final loss is described as follows:

$$\mathscr{L}_{\text{gan}} = \omega_1 \mathscr{L}_{\text{adv}} + \omega_2 \mathscr{L}_{\text{con}} + \omega_3 \mathscr{L}_{\text{enc}}, \tag{2}$$

where the weighting parameters (ω_1, ω_2, and ω_3) are used to modify the effect of individual losses on the overall objective function. Empirically, it has been found that the best values of the parameters are as follows:

$$\omega_a = 1, \omega_b = 1, \omega_1 = 1, \omega_2 = 50, \omega_3 = 1. \tag{3}$$

These results were obtained starting from the relative reference papers of GANomaly [7], where $\omega_1 = 1$, $\omega_2 = 40$, and $\omega_3 = 1$, and DRÆM, where $\omega_a = 1$ and $\omega_b = 1$. Using a branch and bound approach with a step of ± 5 on one ω_* at a time, keeping constant the value of the others. We thus noticed that the weight related to $\mathscr{L}_{\text{con}}$, that is, ω_2, could be increased to 50 with a better result in terms of training time, without losing the contribution of the other components of the main loss.

3.3. Generative-Reconstructive-Discriminative Network with Attention Module.

This work is heavily inspired by DRÆM [6]. This is reflected in the general architecture of the proposed framework. As you can see in Figure 2, the architecture is quite similar to vanilla DRÆM, but we can see the implementation of GANomaly instead of the AE which acted as the reconstructive network. All networks engaged in the reconstructive subnetwork are residual to avoid degradation problems during the

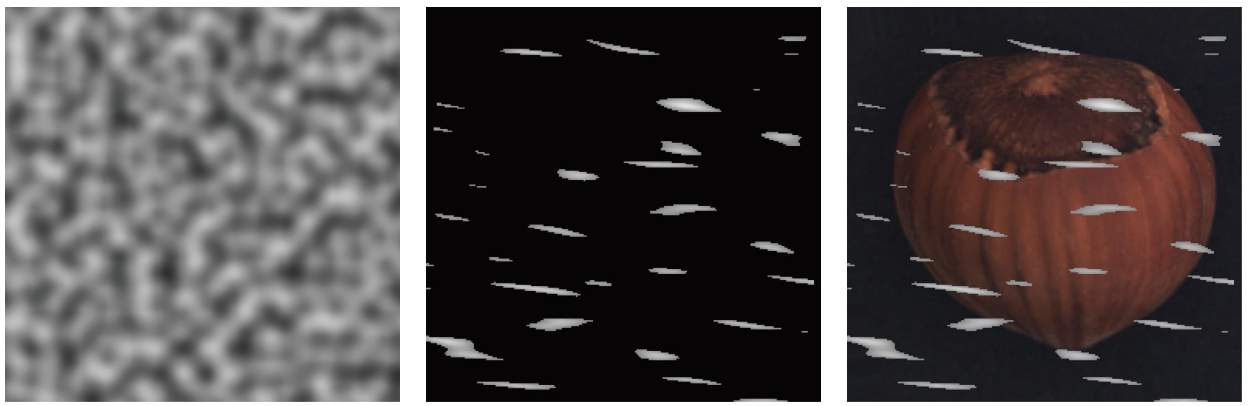


Figure 1: Simulated anomaly generation process: (a) Perlin noise, (b) Perlin noise with random RGB pixels, and (c) dataset's image with Perlin noise.

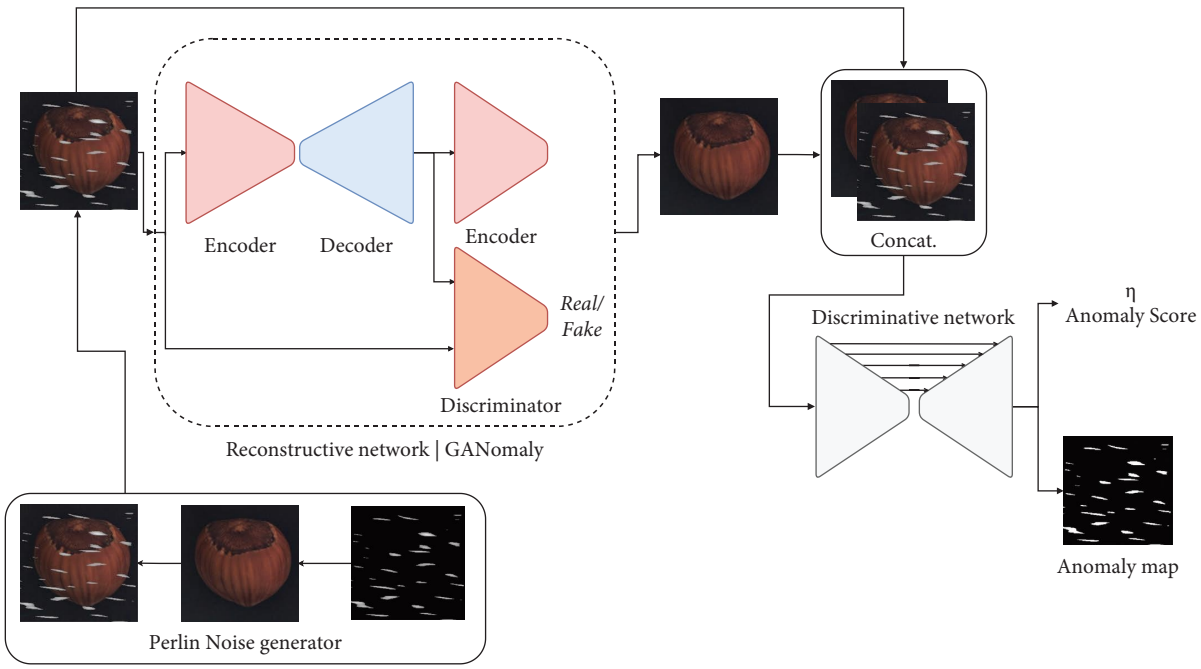


Figure 2: The architecture of DRÆM GAN. The architecture is quite similar to vanilla DRÆM, but we can see the implementation of GANomaly instead of the AE which acted as the reconstructive network.

training. Then, to train the GANomaly engaged in the GRD-Net, the loss described in Equation (2) is used. To train the discriminative network, focal loss (FL) [6, 48] is used. FL can be defined by the following equation:

$$\mathcal{FL}(p) = -(1-p)^{\gamma}\log(p). \tag{4}$$

Basically, $\mathcal{FL}$ adds the factor $-(1-p)^{\gamma}$ to the standard cross entropy. Setting $\gamma > 0$ reduces the relative loss for the well-classified images, putting more focus on the misclassified examples [48]. This loss applied on this subnetwork increases robustness towards accurate segmentation of hard examples. A further improvement was applied to the discriminatory network in order to ensure that only the defects present on the surface of the inspected products are considered. To do this, in addition to the images of the dataset, the network is also given a segmentation mask that highlights the area of interest (AOI) of the product. This mask is multiplied by the anomaly detection mask to obtain an intersection mask. Then, $\mathcal{FL}$ is calculated on this intersection. The overall loss of the GRD-Net became

$$\mathcal{I} = \mathcal{A}_{\text{discr}} \times \mathcal{ROI}_{\text{input}}, \tag{5}$$

where $\mathcal{I}$ is the intersection mask, that is, a tensor obtained by the intersection (multiplication) of the input mask tensor $\mathcal{ROI}_{\text{input}}$ that highlights a ROI (region of interest) in which the network has to segment the anomaly area and the output mask tensor $\mathcal{A}_{\text{discr}}$ of the discriminative network that segments the original image.

$$\mathcal{L}_{\text{tot}} = \mathcal{L}_{\text{gan}} + \mathcal{FL}\left(\mathcal{I}, \mathcal{M}_{\text{input}}\right). \tag{6}$$

So, the total loss function $\mathcal{L}_{\text{tot}}$ is the sum of the GAN loss $\mathcal{L}_{\text{gan}}$ and the focal loss calculated on the intersection area $\mathcal{FL}(\mathcal{I})$.

Finally, the overall training and inference sequences are schematized, respectively, in Figures 3 and 4.

4. Experiments

Several experiments were performed to test the performance of the GRD-Net. First of all, the performances of the GAN with residual convolutional autoencoder have been compared and evaluated with DRÆM and GANomaly, which represent state-of-the-art reconstruction-based anomaly detection and localization technologies. A schema of one stage composed of two residual blocks is shown in Figure 5.

Experiments were conducted over 200 training epochs using vanilla DRÆM autoencoder and our GAN with residual AE. Our network takes inspiration from the ResNet V2 architecture used for the classification task [49].

In this section, we summarize the following:

(1) The residual block applied to our architecture

(2) The two phases of the training loop for the generative part and the discriminative part

(3) The benefits of the residual network applied to autoencoder of the generative part of the GAN

(4) Experiments on the generative-reconstructive-discriminative network (GRD-Net) architecture, with the attention module based on ROIs

(5) A real-case experiments based on pharmaceutical BFS vials, with attention on the body of the aforementioned vials

The first part of the experiments was conducted using three challenging datasets from MVTec's sets: hazelnut, metal nut, and pill datasets. In the second part, the network was tested on hazelnut, zip, and a proprietary pharmaceutical set of BFS strips of vials, from a real study and use case that took place in Bonfiglioli Engineering, for a quality control vision inspection machine. For those datasets, a second ROI dataset was prepared for each training nominal image.

4.1. GAN with Residual AE. As mentioned above, the first experiment aims to challenge the vanilla version of DRÆM architecture. The training lasted 200 epochs, instead of the 700 used for testing DRÆM on the original paper; this provides a more realistic case, which can be implemented on a production line in a real industrial field. During this step, we evaluated the anomaly detection per image performance, using AUROC score, and defect localization within the image, using the AUROC pixelwise score. The learning rate is set to 10^{-4}, and we used a policy based on "reduction on plateau" heuristic with a patience of 3 epochs and a reduction factor $\alpha = 0.1$. When a plateau of 3 epochs is reached at epoch k, it decreases using the formula:

$$\mathcal{LR}_k = \mathcal{LR}_{k-1} \cdot e^{-\alpha}, \tag{7}$$

where $\mathcal{LR}_k$ is the learning rate at the k-th epoch.

For the evaluation, we used the AUROC, widely used in architecture comparisons, at the image level and at the pixel level, as semisupervised anomaly detection and localization score.

Data augmentation is performed on training examples, using a random rotation in the range of $[-\pi/2, +\pi/2]$ radians in order to reduce overfitting during training over lots of epochs because of the small number of anomaly-free images provided in MVTec datasets.

For the sake of completeness, we also tested and compared the GRD-Net with a vanilla convolutional autoencoder (that is without residual block) and the GRD-Net with fully convolutional residual autoencoder.

The experiment was performed using our huge pharmaceutical dataset on 500 epochs, using only the generative part, comparing the losses. Because this is a second network, the disciminative one that segments the defect within the image depends strictly on the performance of the first, and its architecture has not been modified.

The experimental results are very encouraging in support of the intuition that the residual network, even in the case of an autoencoder applied to a GAN, is more effective in generating the final data.

This can be visually appreciated in Figure 6. We also provided a comparison between losses used for the generator in Table 1.

4.1.1. Anomaly Detection. For what concerns surface anomaly detection, our proposed architecture enhances somewhat not only the final score of the two reference models but improves also the learning curve making it smoother and steeper toward convergence, especially during the first transitional period. In addition to this, also the difference between training and validation curves is far less with our model. The smoothing of the learning curve can be explained by the GAN model that, with the discriminator network, improves the stability of the training process. The steepest incline and a lower presence of the overfitting phenomenon (that can be observed with the higher difference between validation and training curves), can be attributed to both the GAN model and residual network. This is due to the improvement given by the adversarial part of the GAN and by the reduction of the gradient vanishing that could affect deep convolutional networks.

4.1.2. Anomaly Localization. As for anomaly detection, also anomaly localization has been compared with DRÆM after 200 epochs of training. GANomaly was not included in this comparison because it does not exist in the official paper, a method capable of locating defective regions. Tables 2–5 show the AUROC result comparison between DRÆM and our approach, as mentioned above in four different stages of the training phase: after 10, 50, 100, and 200 epochs. The results are very encouraging because they improve those of the vanilla network; in fact, a better quality of the

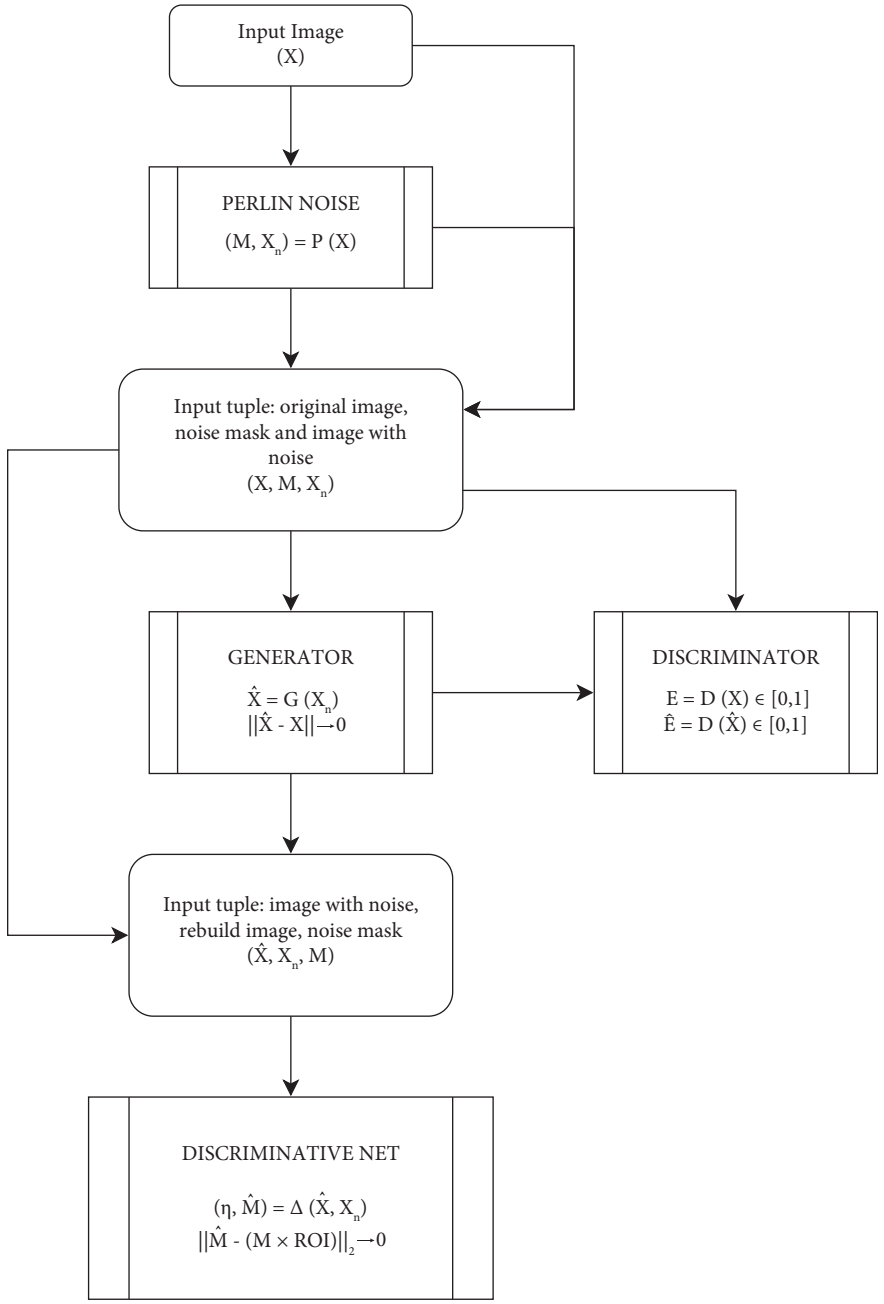


Figure 3: Train step flowchart: input image X is transformed in X_n, that is, the image with the Perlin noise superimposed. M is the mask image of the noise areas.

reconstructed image implies a better performance of the second discriminative network. Moreover, as in Figure 7, it is clear how validation curves are much better in our (red) model, compared to the vanilla one (orange), especially in the first phase of the training, thus reducing the number of needed epochs for obtaining an acceptable result for an industrial process.

On the other hand, embedding similarity-based network, such as PatchCore, seems to have a better pixelwise AUROC score. But because of the nature itself of the architecture, it is not possible to add, in an easy way, an attention module based on ROIs.

4.2. GRD-Net with ROI. In the second experiment, the network's capability to learn an interesting region was tested in which and only within it anomalies can be spotted and located. This region of interest is arbitrarily defined in the training set, within the input image area. Zipper and Hazelnut datasets were used for the purpose. Especially, Zipper is particularly suitable, since samples have 2 logic regions of interest: the zipper area itself and the fabric area. In our case, we used as the region of interest of the zipper part, so we would exclude defects on the fabric zone. An example is shown in Figure 8. As previously explained in Section 3.3, the discriminative network was trained using as

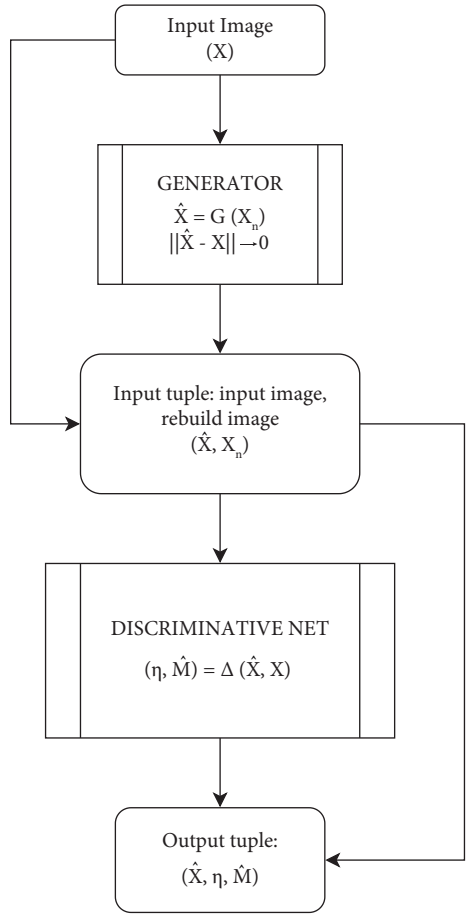


Figure 4: Inference step flowchart.

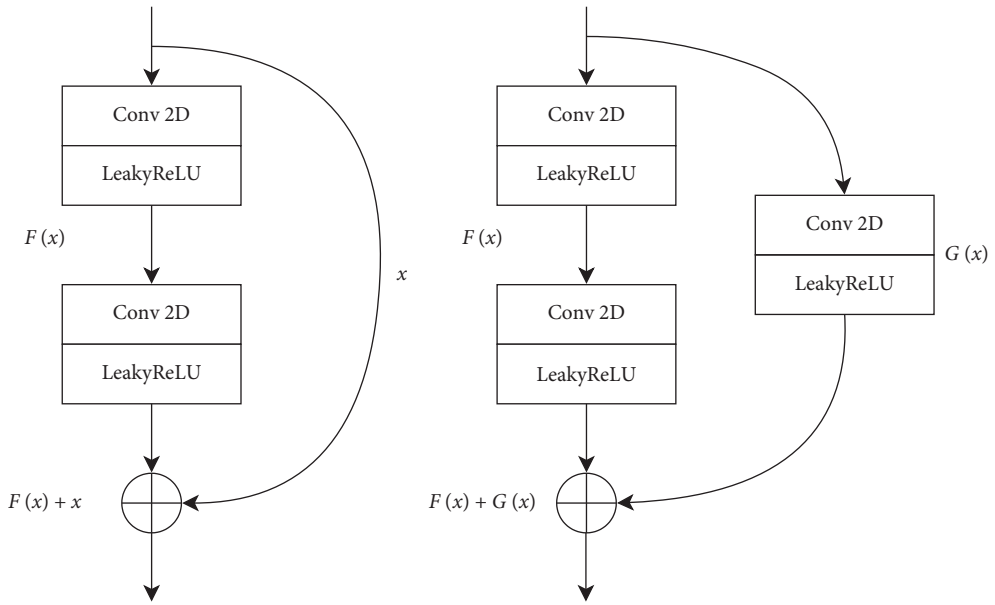


Figure 5: Two consecutive residual blocks of one stage of the encoder network. The introduction of a residual architecture in the encoder-decoder-encoder GAN [49] revealed to be more stable during the training phase, by giving better results with equal epochs.

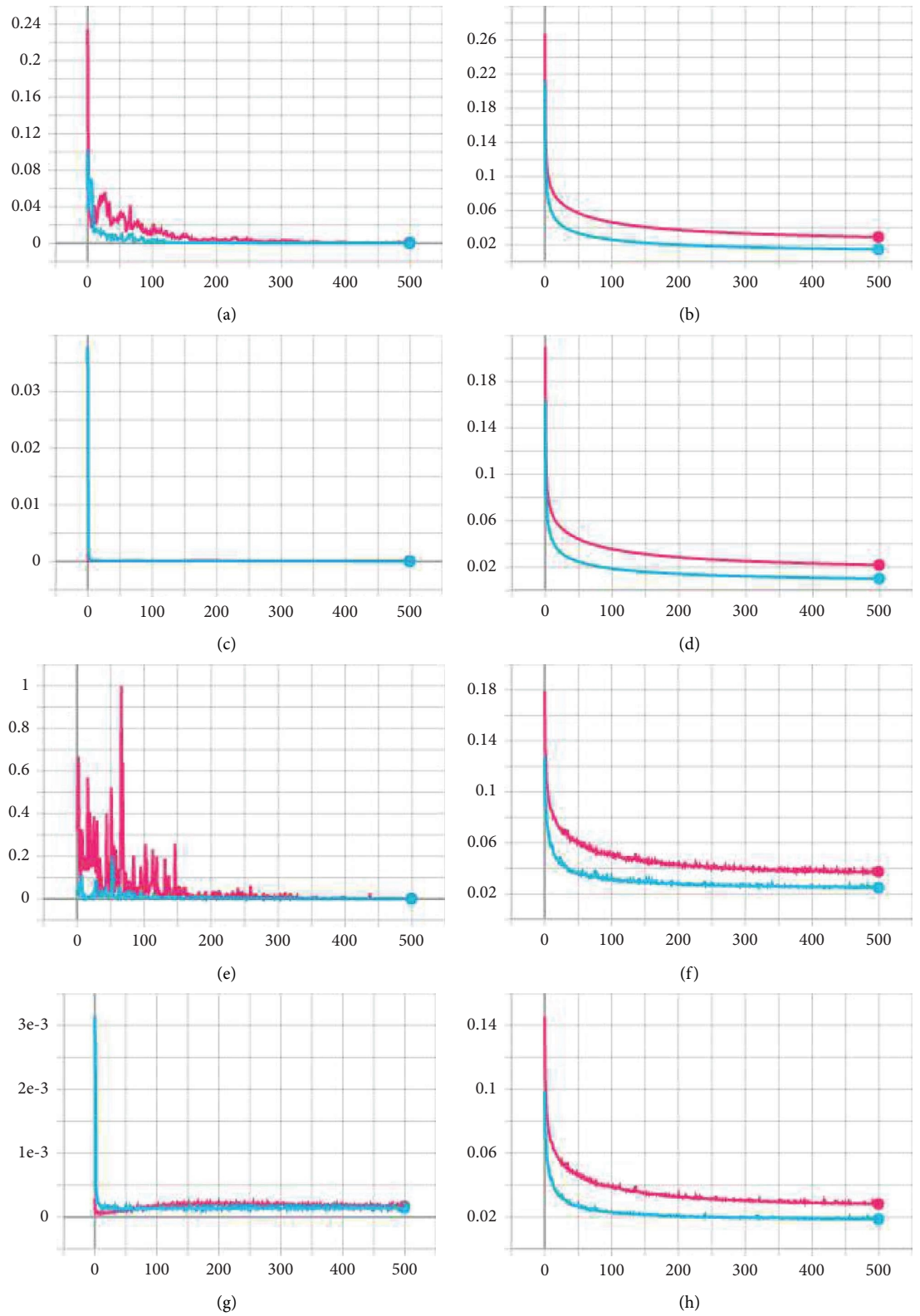


Figure 6: Visual representation of how the network with vanilla autoencoder (magenta) is not only less effective, but also noisier in some losses, such as adversarial loss, compared to the residual architecture (cyan): (a) training adversarial loss, (b) training contextual loss, (c) training encoder loss, (d) training SSIM loss, (e) validation adversarial loss, (f) validation contextual loss, (g) validation encoder loss, and (h) validation SSIM loss.

Table 1: Comparison between vanilla and residual architecture used for generative part of the GAN architecture in the GRD-Net.

	Loss value for training (validation) phase			
	Epoch 250 vanilla	Epoch 250 residual	Epoch 500 vanilla	Epoch 500 residual
Adversarial loss	2.8811×10^{-3} (5.1180×10^{-2})	3.2498×10^{-4} (3.8928×10^{-3})	7.4750×10^{-4} (1.3720×10^{-3})	1.3797×10^{-4} (4.5011×10^{-4})
Contextual loss	0.03502 (0.04136)	0.01853 (0.02782)	0.02912 (0.03747)	0.01460 (0.02488)
Encoder loss	1.7221×10^{-4} (2.1078×10^{-4})	6.8905×10^{-5} (1.2333×10^{-4})	1.1745×10^{-4} (1.6965×10^{-4})	4.6982×10^{-5} (1.4363×10^{-4})
SSIM loss	0.02665 (0.03115)	0.01304 (0.02010)	0.02200 (0.02830)	0.01014 (0.01872)

Table 2: AUROC score after 10 epochs of training per image (pixel).

	AUROC per image (pixel) at 10 epochs	
	DRÆM	GRD-Net
Hazelnut	73.1 (55.7)	96.7 (91.0)
Metal nut	58.3 (49.0)	96.4 (69.3)
Pill	74.2 (66.0)	77.5 (90.5)

Table 3: AUROC score after 35 epochs of training per image (pixel).

	AUROC per image (pixel) at 35 epochs	
	DRÆM	GRD-Net
Hazelnut	85.3 (82.4)	99.5 (95.5)
Metal nut	61.8 (49.0)	99.3 (69.3)
Pill	75.7 (86.5)	89.8 (95.5)

Table 4: AUROC score after 100 epochs of training per image (pixel).

	AUROC per image (pixel) at 100 epochs	
	DRÆM	GRD-Net
Hazelnut	98.8 (94.8)	100.0 (97.3)
Metal nut	99.7 (86.7)	99.8 (70.4)
Pill	93.8 (94.8)	98.2 (95.5)

focal loss variable intersection between ROI and the mask generated from Perlin noise. In this way, the net will start generalizing not only how to spot anomalies from differences between original and reconstructed images but also where the region of interest is and in which to focus the search for differences. In fact, in most industrial cases, the totality of the image is not important; indeed, sometimes it could be misleading, as there may be anomalies within the frame that are not part of the product itself.

Table 5: Final comparative table with AUROC score between GANomaly (200 epochs), DRÆM (200 epochs), PaDiM (ResNet18 pretrain), PatchCore (ResNet50 pretrain,) and GRD-Net (200 epochs).

	AUROC per image (pixel)				
	GANomaly	DRÆM	PaDiM	PatchCore	GRD-Net
Hazelnut	78.5 (—)	100.0 (95.0)	— 97.7	100.0 98.6	100.0 (97.4)
Metal nut	70.0 (—)	98.7 (86.7)	— 96.7	99.7 98.4	100.0 (96.2)
Pill	74.3 (—)	97.9 (94.8)	— 94.7	97.0 97.1	98.5 (95.8)
Cable	75.7 (—)	91.8 (94.7)	— 96.7	99.3 98.2	99.5 (98.1)

The results of GANomaly and DRÆM are obtained by us adjusting the number of training epochs to the number of training epochs used to train the GRD-Net; for this reason, the final result may vary a little from the reference paper.

In order to obtain this result, a ROI for each training image was created and the focal loss was customized as explained in Section 3.3, namely, by intersecting mask during training and the aforementioned region of interest. Thus, the discriminative network learns to generalize the most important part of the image, where to focus the attention.

4.3. Real-Case Experiment. The studied model was used in a real-case industrial process to perform a quality control on pharmaceutical BFS strips of vials. Tests are performed by a Bonfiglioli Engineering automatic machine, with a rotary carousel with a tracker where acquiring sensors are installed. The training set is composed by 230355 images of vials, acquired in 3 different areas by an online camera, during the production process. For reasons related to nondisclosure agreements, we cannot show full product images, but only a limited area that covers one of the most interesting parts of our aim. Strips consist of 5 BFS plastic vials, which stick to each other on the long side and liquid filled. Because of these features, one of the most challenging areas is the meniscus

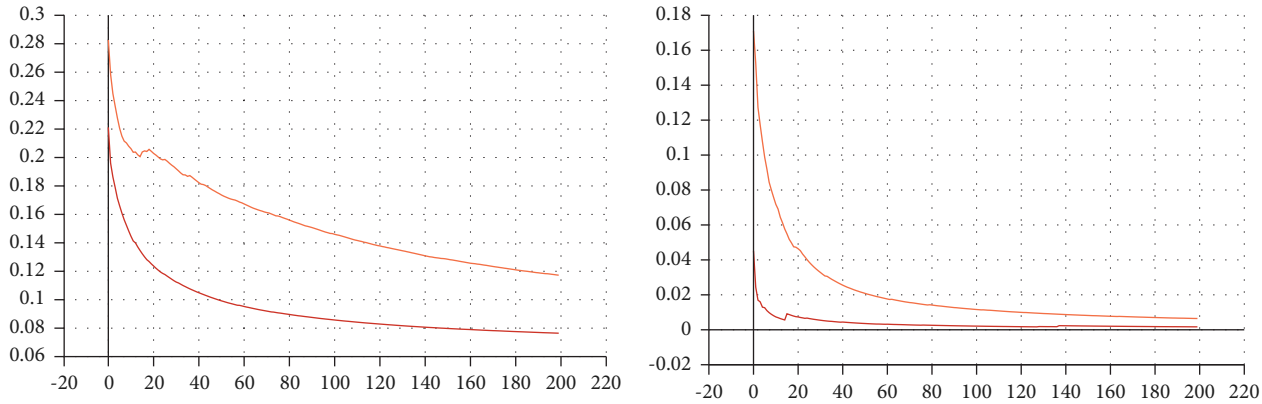


Figure 7: Validation losses for generative (a) and discriminative (b) subnetworks. Red curve is obtained during training of our model, the orange one is obtained with the vanilla model. It is evident that the learning curve is much better in our case, for both nets: (a) validation contextual loss and (b) validation focal loss.

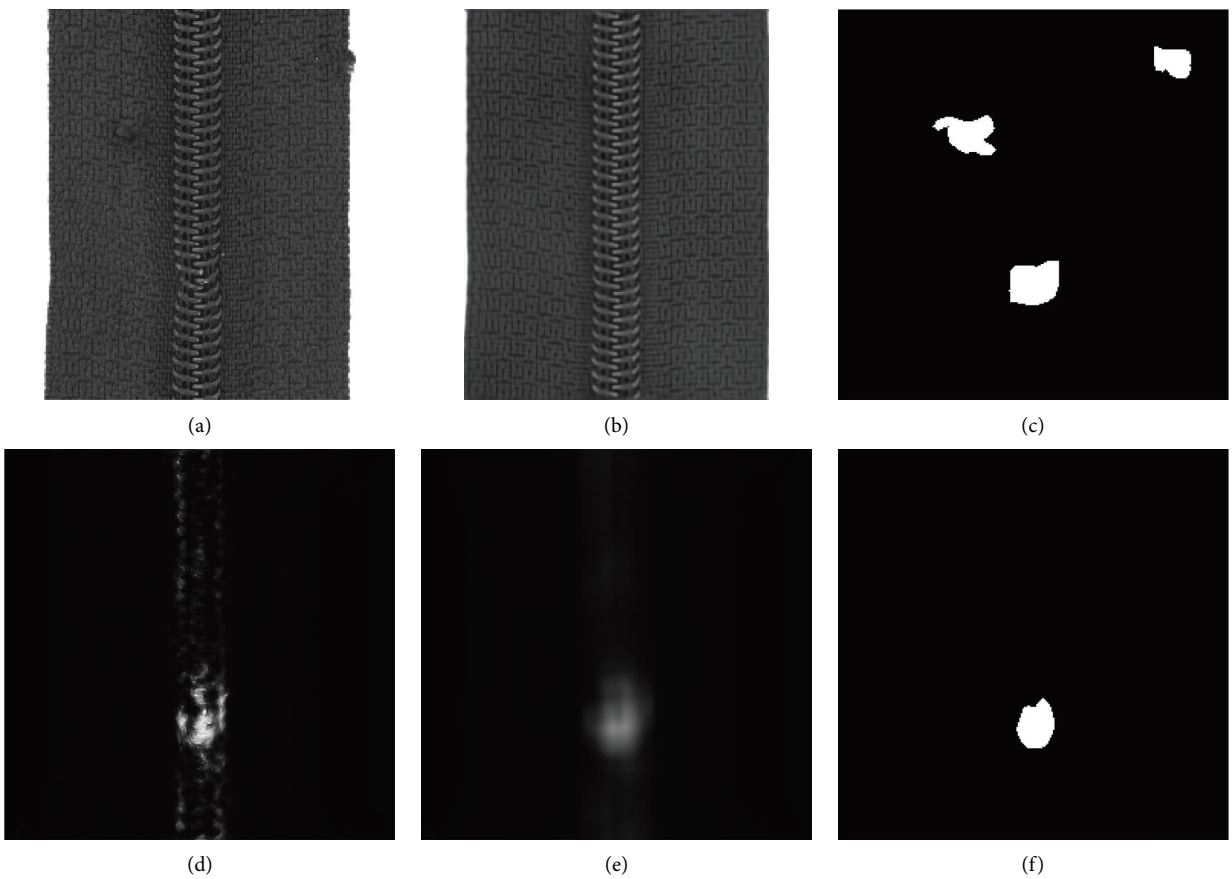
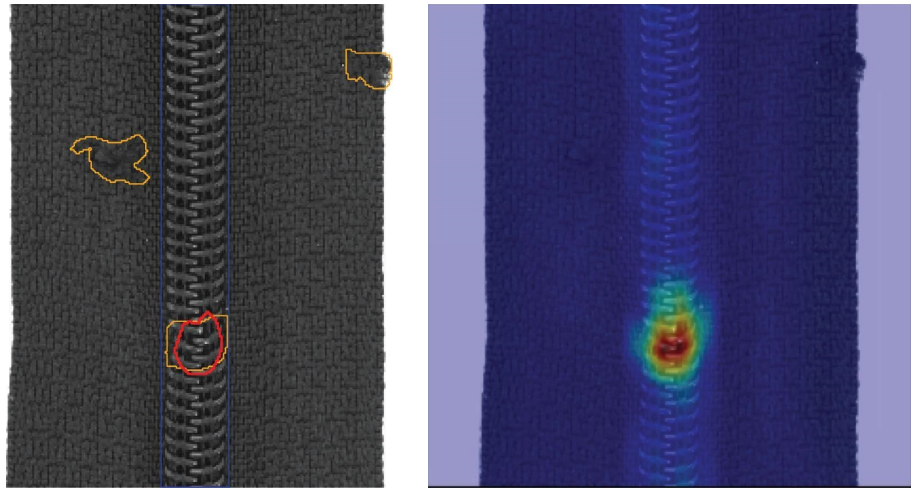


Figure 8: Continued.



(g) (h)

Figure 8: Especially, significant example from the zipper dataset in which we could spot 3 anomalies: one in the zipper, one in the middle of fabric part, and another, the last, on the border of the fabric zone. As we can see, the only spotted is the one in the zipper region, perfectly inside ROI, and almost perfectly aligned with the ground truth defect region: (a) original image (X), (b) reconstructed image by the generator $G(\tilde{X})$, (c) ground truth (M), (d) generated heatmap by discriminative model, (e) generated heatmap by discriminative model after average pooling with 21×21 kernel, (f) result generated anomaly localization region $(\tilde{M})$, (g) original image with regions: blue region is the ROI, the orange region is the ground truth (M), and finally the red region is the generated region $(\tilde{M})$, and (h) image generated overimposing the convoluted heatmap from the discriminative net to X, and colorizing it with jet color-map.

(a) (b) (c)

Figure 9: 3 examples of real cases where algorithmic analysis is difficult, if not almost impossible: (a) floating black particle on meniscus, near the shoulder of the vial, (b) black spot near the meniscus, and (c) scratch at the turn of horizontal engraving.

region. This is due to the great randomness and variability of the aforementioned meniscus. Its own shape and the possibility that there could be bubbles under it or liquid drops over it make it very difficult to treat this region using only classical blob analysis algorithms.

Figure 9 shows 3 real-case examples where blob analysis is almost impossible due to the variability of the meniscus shape and the shadows generated by the shape of the product itself and the position of the sensor in relation of the product.

With our network, we managed to localize those anomalies, with good result, acceptable compared to human and classical algorithms' scores. These results could be seen in Figure 10 and in Table 6.

4.4. Ablation Study. The GRD-Net architecture is analyzed, evaluating the network generative model and the loss of the discriminative part.

4.4.1. Generative Model. The generative subnet, namely, the reconstructive part, was challenged starting from the SoA described in the DRÆM paper [6], in 4.2. *Ablation Study-Architecture* section. Adding the GAN structure with a residual autoencoder, the latter has been tested using a full-convolutional bottleneck, with a latent size of $z = 32 \times 8 \times 8$, and a dense bottleneck, with a latent size of $z = 2048$. As previously shown, best performance was obtained using our

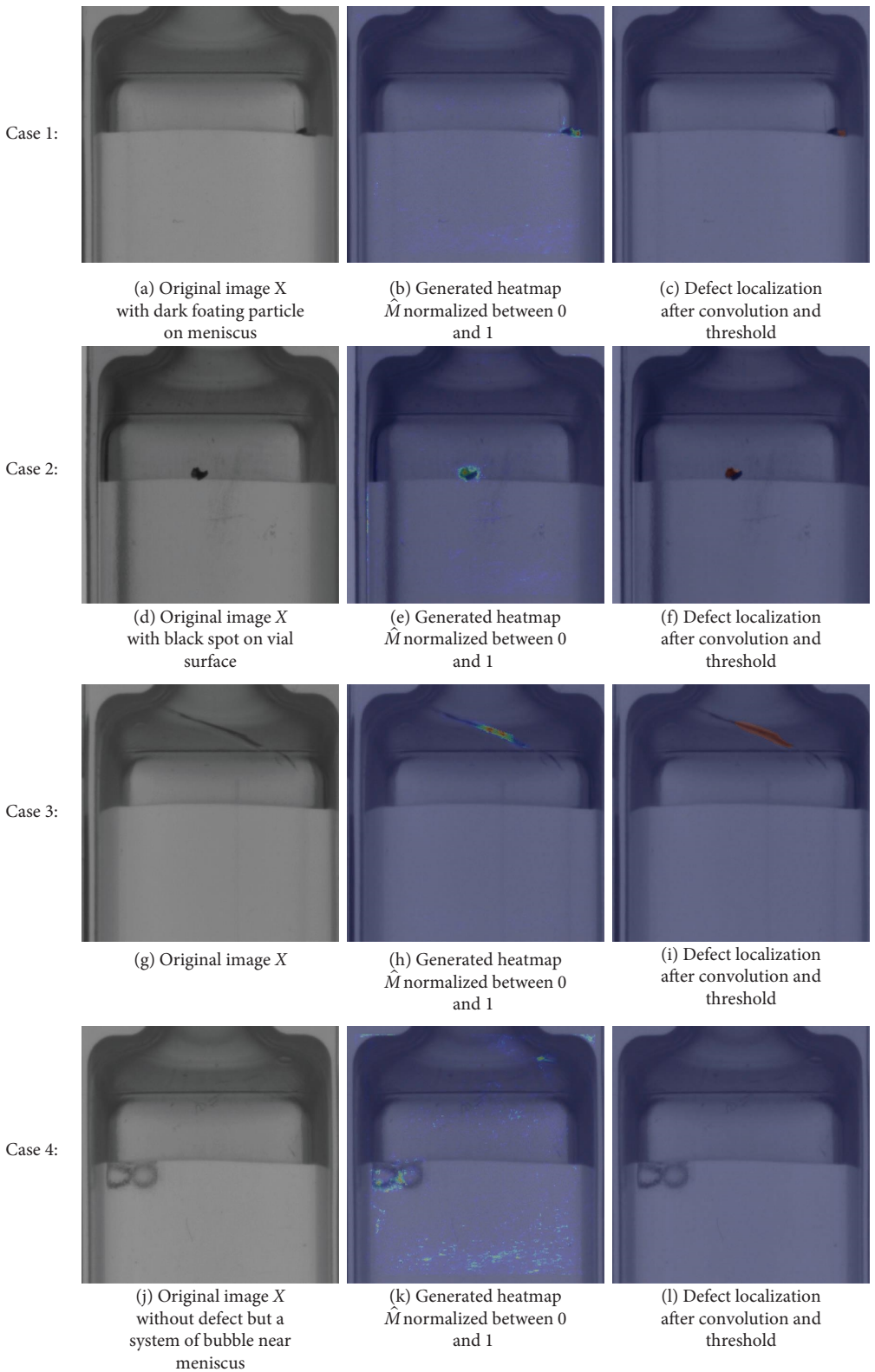


Figure 10: Visual results on real-case experiment. The first 3 images represent a defect and the last a regular product really difficult to spot.

GAN architecture, with a fully convolutional residual autoencoder (CRAE). Dense-bottleneck residual autoencoder (DRAE), on the other hand, is a good alternative, and in some cases, it is better in anomaly removal task but is less capable of learning the aleatory areas. A good example, shown in Figure 11, is a pill dataset, whose pills, used as examples, have a random-like dotted reddish texture that is better reproduced with a fully convolutional bottleneck.

Table 6: Real-case experiment statistics after 30 epochs of training.

	Best results on 30 epochs training		
	Best AUROC per image	Best AUROC per pixel	Best accuracy
Vials on the meniscus area	0.981	0.996	0.932

(a) (b) (c) (d)

Figure 11: (a) Original pill image (X), (b) pill image with Perlin noise $X_{(n)}$ ($\widehat{X}$), (c) pill image rebuilt by the GRD-Net with CRAE ($\widehat{X}$), and (d) pill image rebuilt by the GRD-Net with DRAE ($\widehat{X}$).

4.4.2. Discriminative Model Loss. Discriminative model loss was originally composed by focal loss added to the cross entropy overlap distance loss [50, 51]. The initial idea was that the second addendum would help to focus attention of the network only into the ROI area. So, the first idea was

$$\mathscr{L}_{\text{overlap}}\left(\mathscr{A}_{\text{discr}}, \mathscr{ROI}_{\text{input}}\right) = w\left(1 - \frac{\left|\mathscr{A}_{\text{discr}} \cap \mathscr{ROI}_{\text{input}}\right|}{\min\left(\left|\mathscr{A}_{\text{discr}}, \mathscr{ROI}_{\text{input}}\right|\right)}\right), \tag{8}$$

with $w \in [0, 1]$, where $\mathscr{L}_{\text{overlap}}$ is the contribution of the cross entropy overlap distance loss in the discriminative loss, w is a hyperparameter, $\mathscr{A}_{\text{discr}}$ is the area mask generated by the discriminative network, and $\mathscr{ROI}_{\text{input}}$ is the reference ROI.

$$\mathscr{L}_{FL} = \mathscr{FL}\left(\mathscr{A}_{\text{discr}}, \mathscr{M}_{\text{input}}\right) + \mathscr{L}_{\text{overlap}}\left(\mathscr{A}_{\text{discr}}, \mathscr{ROI}_{\text{input}}\right). \tag{9}$$

This loss led the discriminative network to focus on the ROI, but also led to highlight all the ROI area on the heatmap generated as discriminative net output. This is because (8) meant that the $\mathscr{A}_{\text{discr}}$ region tends to $\mathscr{ROI}_{\text{input}} \cdot w$. In order to prevent this issue, we performed 4 experiments, with 4 different variations of the $\mathscr{L}_{\text{FL}}$:

(1) For the first experiment, we used (9)

(2) For the second trial, we used the vanilla focal loss, but with the intersection, as in equation (5), $\mathscr{I} = |\mathscr{A}_{\text{discr}} \cap \mathscr{ROI}_{\text{input}}| = \mathscr{A}_{\text{discr}} \times \mathscr{ROI}_{\text{input}}$, as the focal loss function input

(3) For the third experiment, we added to the vanilla loss with the input explained in the previous point, the overlap custom loss

(4) For the fourth, and last, test, we negated the overlap function to not intersect $\mathbf{1} - \mathscr{ROI}_{\text{input}}$

Best results, both visually (as shown in Figure 12) and numerically (as shown in Table 7), were obtained using method 2. This is due to the tendency to carry $\min(\mathscr{A}_{\text{discr}} \cap \mathscr{ROI}_{\text{input}})$ to be w. Similar results were obtained on the zipper dataset that was a good benchmark for real-case defects that, on the same image, appear both inside and outside the ROI, as illustrated in Figure 8.

5. Conclusions

The aim of this work is to create an anomaly detection network that pays attention mainly to a specific part of an image to avoid the identification of part of images containing noise defects in the background. This new architecture called generative-reconstructive-discriminative anomaly detection with the region of interest attention module network (GRD-Net) is based on two state-of-the-art anomaly detection networks: GANomaly and DRÆM. GDR-Net is composed by a first generative-reconstructive part (GANomaly) trained to identify and reconstruct anomalies, maintaining the nonanomalous regions of the input image. The first submodel maps the input image to a lower dimension vector

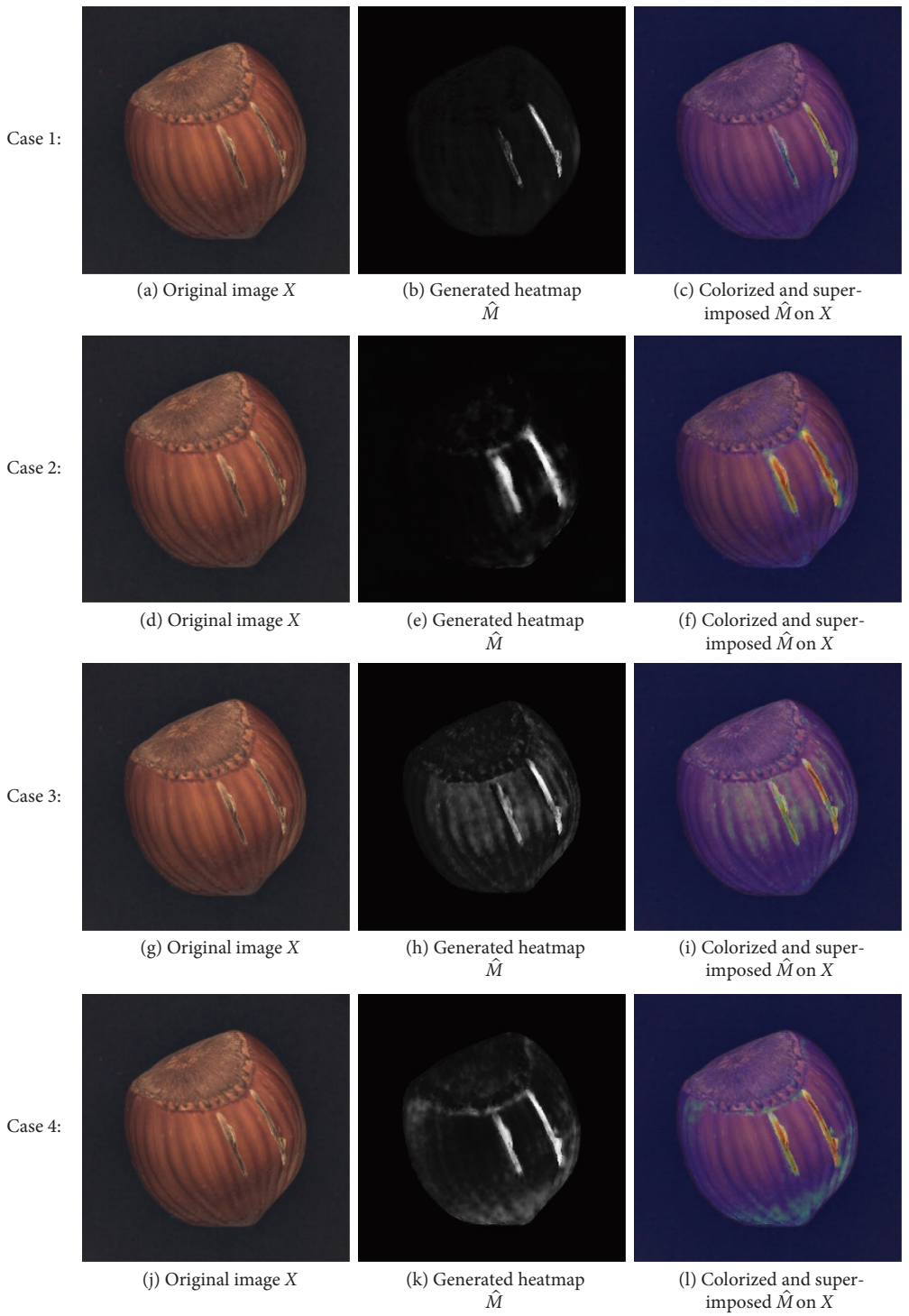


Figure 12: Visual comparison between 4 losses, for the discriminative network. The second case is more clear that segment better only the anomalous areas.

Table 7: AUROC score after 200 epochs of training per image (pixel).

	AUROC per image (pixel) at 100 epochs		
	AUROC	AUROC pixel	Accuracy
Case 1	99.4	94.3	94.6
Case 2	**100.0**	**95.3**	**100.0**
Case 3	99.9	91.6	99.1
Case 4	**100.0**	93.5	**100.0**

Bolded value are higher values among the cases.

using encoder-decoder-encoder subnetworks, which are then used to reconstruct the generated output image. In order to learn joint anomaly inclusion reconstruction and create accurate anomaly segmentation maps, the second network combines the original and reconstructed images. In order to ensure that only the defects present on the surface of the inspected products are considered; in addition to the images of the dataset, the network is also given a segmentation mask that highlights the area of interest (AOI) of the product. This mask is multiplied by the anomaly detection mask generated by the discriminative network to obtain an intersection mask. This contribution is summed to the loss of the network. The GRD-Net was tested on all MVTec-AD datasets, on an updated version of the zipper MVTec-AD dataset and on a real industrial dataset provided by company Bonfiglioli Engineering, located in Ferrara (IT). Experiments show that the GRD-Net performs better than both DRÆM and GANomaly not only in terms of performance (AUROC) but also in visual terms. In fact, the experiments show that the attention module allows the GRD-Net to identify as real defects only to those that are in the AOI of the product. In this way, the noise introduced by random variations in the background makes no negative contribution to the performance and reliability of the system created.

Nomenclature

AE:	Autoencoder
VAE:	Variational autoencoder
CNN:	Convolutional neural network
RNN:	Recurrent neural network
LSTM:	Long short-term memory
GAN:	Generative adversarial network
Generator:	Generative subnet of the GAN
Discriminator:	Adversarial subnet of the GAN
Discriminative net:	U-Net subsequent to the GAN used for segmentation
CRAE:	Convolutional residual autoencoder
DRAE:	Dense-bottleneck residual autoencoder
AUROC:	Area under the receiver operating characteristic
ROI:	Region of interest
SSIM:	Structural similarity index measure.

Data Availability

The MVTec dataset data used to test the software and models of this study can be found on the proprietary website, https://www.mvtec.com/company/research/datasets/mvtec-ad. The software and real-case dataset data used to build the architecture and test the model of this study are restricted by the Bonfiglioli Engineering NDA in order to protect client data. Data are available from Niccolò Ferrari (niccolo.ferrari@unife.it) for researchers who meet the criteria for accessing confidential data.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank Bonfiglioli Engineering for providing a real-case dataset to test the software developed in this work. The first author was supported by an industrial PhD funded by Bonfiglioli Engineering, Ferrara, Italy. The other author was supported by a PhD scholarship funded by the Emilia-Romagna region, Italy, under POR FSE 2014–2020 program.

References

[1] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," 2020, https://arxiv.org/abs/2003.05991.

[2] U. Michelucci, "An introduction to autoencoders," 2022, https://arxiv.org/abs/2201.03898.

[3] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," 2019, https://arxiv.org/abs/1906.02691.

[4] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2013, https://arxiv.org/abs/1312.6114.

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[6] V. Zavrtanik, M. Kristan, and D. Skočaj, "Draem-a discriminatively trained reconstruction embedding for surface anomaly detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8330–8339, Montreal, BC, Canada, October 2021.

[7] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "Ganomaly: semi-supervised anomaly detection via adversarial training," in *Asian Conference on Computer Vision*, pp. 622–637, Springer, Berlin, Germany, 2018.

[8] C. Wickramasinghe, D. Marino, and M. Manic, "resnet autoencoders for unsupervised feature learning from high-dimensional data: deep models resistant to performance degradation," *IEEE Access*, vol. 9, p. 1, 2021.

[9] S. Akçay, A. Atapour-Abarghouei, and T. P. Breckon, "Skip-ganomaly: skip connected and adversarially trained encoder-decoder anomaly detection," in *Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, Budapest, Hungary, July 2019.

[10] B. Paul, S. Löwe, M. Fauser, D. Sattlegger, and C. Steger, "Improving unsupervised defect segmentation by applying structural similarity to autoencoders," 2018, https://arxiv.org/abs/1807.02011.

[11] D. Gong, L. Liu, V. Le et al., "Memorizing normality to detect anomaly: memory-augmented deep autoencoder for unsupervised anomaly detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1705–1714, Montreal, BC, Canada, June 2019.

[12] S. Venkataramanan, K.-C. Peng, R. V. Singh, and A. Mahalanobis, "Attention guided anomaly localization in images," in *European Conference on Computer Vision*, pp. 485–503, Springer, Berlin, Germany, 2020.

[13] S. Pidhorskyi, R. Almohsen, and G. Doretto, "Generative probabilistic novelty detection with adversarial autoencoders," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[14] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, "Adversarially learned one-class classifier for novelty detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3379–3388, Vancouver, BC, Canada, June 2018.

[15] X. Xia, X. Pan, N. Li et al., "Gan-based anomaly detection: a review," *Neurocomputing*, vol. 493, pp. 497–535, 2022.

[16] B. Paul, M. Fauser, D. Sattlegger, and C. Steger, "Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9592–9600, Long Beach, CA, USA, June 2019.

[17] X. Xie, Y. Huang, W. Ning, D. Wu, Z. Li, and H. Yang, "Rdad: a reconstructive and discriminative anomaly detection model based on transformer," *International Journal of Intelligent Systems*, vol. 37, no. 11, pp. 8928–8946, 2022.

[18] P. Perera, N. Ramesh, and B. Xiang, "Ocgan: one-class novelty detection using gans with constrained latent representations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2898–2906, Baltimore, MD, USA, June 2019.

[19] O. Rippel, P. Mertens, and D. Merhof, "Modeling the distribution of normal data in pre-trained deep features for anomaly detection," in *Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 6726–6733, IEEE, Milan, Italy, January 2021.

[20] L. Bergman, N. Cohen, and Y. Hoshen, "Deep nearest neighbor anomaly detection," 2020, https://arxiv.org/abs/2002.10445.

[21] P. Napoletano, F. Piccoli, and R. Schettini, "Anomaly detection in nanofibrous materials by cnn-based self-similarity," *Sensors*, vol. 18, no. 2, p. 209, 2018.

[22] N. Cohen and Y. Hoshen, "Sub-image anomaly detection with deep pyramid correspondences," 2020, https://arxiv.org/abs/2005.02357.

[23] T. Defard, A. Setkov, A. Loesch, and R. Audigier, "Padim: a patch distribution modeling framework for anomaly detection and localization," in *Proceedings of the International Conference on Pattern Recognition*, pp. 475–489, Springer, Berlin, Germany, March 2021.

[24] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. Gehler, "Towards total recall in industrial anomaly detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14318–14328, New Orleans, LA, USA, June 2022.

[25] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real nvp," 2016, https://arxiv.org/abs/1605.08803.

[26] Y. Shi, J. Yang, and Z. Qi, "Unsupervised anomaly segmentation via deep feature reconstruction," *Neurocomputing*, vol. 424, pp. 9–22, 2021.

[27] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *International conference on information processing in medical imaging Lecture Notes in Computer Science*, M. Niethammer, Ed., vol. 10265, pp. 146–157, Springer, Berlin, Germany, 2017.

[28] T. Schlegl, P. Seeböck, S. M. Waldstein, G. Langs, and U. Schmidt-Erfurth, "f-AnoGAN: fast unsupervised anomaly detection with generative adversarial networks," *Medical Image Analysis*, vol. 54, pp. 30–44, 2019.

[29] M. Rudolph, B. Wandt, and B. Rosenhahn, "Same same but differnet: semi-supervised defect detection with normalizing flows," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 1907–1916, Hanover, Germany, Augest 2021.

[30] D. Gudovskiy, S. Ishizaka, and K. 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*, pp. 98–107, Tokyo, Japan, July 2022.

[31] J. Bae, J.-H. Lee, and S. Kim, "Pni: industrial anomaly detection using position and neighborhood information," 2023, https://arxiv.org/abs/2211.12634.

[32] B. Paul, M. Fauser, D. Sattlegger, and C. Steger, "Uninformed students: student-teacher anomaly detection with discriminative latent embeddings," 2019, https://arxiv.org/abs/1911.02357.

[33] P. Bergmann, K. Batzner, M. Fauser, D. Sattlegger, and C. Steger, "Beyond dents and scratches: logical constraints in unsupervised anomaly detection and localization," *International Journal of Computer Vision*, vol. 130, no. 4, pp. 947–969, 2022.

[34] K. Batzner, L. Heckler, and R. König, "Efficientad: accurate visual anomaly detection at millisecond-level latencies," 2023, https://arxiv.org/abs/2303.14535.

[35] D. Li, D. Chen, L. Shi, B. Jin, J. Goh, and S.-K. Ng, "Mad-gan: multivariate anomaly detection for time series data with generative adversarial networks," 2019, https://arxiv.org/abs/1901.04997.

[36] G. Zhu, H. Zhao, H. Liu, and H. Sun, "A novel lstm-gan algorithm for time series anomaly detection," in *2019 Prognostics and System Health Management Conference (PHM-Qingdao)*, pp. 1–6, Qingdao, China, October 2019.

[37] Z. Niu, K. Yu, and X. Wu, "Lstm-based vae-gan for time-series anomaly detection," *Sensors*, vol. 20, no. 13, p. 3738, 2020.

[38] W. Jiang, Y. Hong, B. Zhou, X. He, and C. Cheng, "A gan-based anomaly detection approach for imbalanced industrial time series," *IEEE Access*, vol. 7, pp. 143608–143619, 2019.

[39] P. Radoglou Grammatikis, P. Sarigiannidis, G. Efstathopoulos, and E. Panaousis, "Aries: a novel multivariate intrusion detection system for smart grid," *Sensors*, vol. 20, no. 18, p. 5305, 2020.

[40] I. Siniosoglou, P. Radoglou-Grammatikis, G. Efstathopoulos, P. Fouliras, and P. Sarigiannidis, "A unified deep learning anomaly detection and classification approach for smart grid environments," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1137–1151, 2021.

[41] D. Chen, L. Yue, X. Chang, M. Xu, and T. Jia, "Nm-gan: noise-modulated generative adversarial network for video anomaly detection," *Pattern Recognition*, vol. 116, Article ID 107969, 2021.

[42] K. Perlin, "An image synthesizer," *ACM Siggraph Computer Graphics*, vol. 19, no. 3, pp. 287–296, 1985.

[43] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[44] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, Cambridge, MA, USA, 2016.

[45] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: an overview," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, 2018.

[46] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," 2015, https://arxiv.org/abs/1511.05644.

[47] M. Mirza and O. Simon, "Conditional generative adversarial nets," 2014, https://arxiv.org/abs/1411.1784.

[48] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 2980–2988, 2017.

[49] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015, https://arxiv.org/abs/1512.03385.

[50] M. Fraccaroli, A. Bizzarri, P. Casellati, and E. Lamma, "Exploiting cnn's visual explanations to drive anomaly detection," *Submitted to Applied Intelligence*, Springer, Berlin, Germany, 2021.

[51] M. Fraccaroli, A. Bizzarri, P. Casellati, and E. Lamma, "Cross entropy overlap distance," 2022, https://ml.unife.it/wp-content/uploads/Papers/FraBizCasLam-ITAL_IA22.pdf.