Journal Pre-proof

Reconstruction by inpainting for visual anomaly detection

Vitjan Zavrtanik , Matej Kristan, Danijel Skčaj

PII: S0031-3203(20)30509-4
DOI: https://doi.org/10.1016/j.patcog.2020.107706
Reference: PR 107706

To appear in: *Pattern Recognition*

Received date: 29 May 2020
Revised date: 22 September 2020
Accepted date: 14 October 2020

Please cite this article as: Vitjan Zavrtanik , Matej Kristan, Danijel Skčaj, Reconstruction by inpainting for visual anomaly detection, *Pattern Recognition* (2020), doi: https://doi.org/10.1016/j.patcog.2020.107706

Highlights:

- A reconstruction-by-inpainting-based anomaly detection method (RIAD) was proposed.
- RIAD achieves state-of-the-art performance on anomaly detection and localization.
- We compare RIAD anomaly detection results with recent anomaly detection methods.
- The generality of RIAD is demonstrated by applying it on video anomaly detection.

Reconstruction by inpainting for visual anomaly detection

Vitjan Zavrtanik[a], Matej Kristan[a], Danijel Skoaj[a]

[a]*Faculty of Computer and Information Science, University of Ljubljana, Vena pot 113, 1000 Ljubljana, Slovenia*

Abstract

Visual anomaly detection addresses the problem of classification or localization of regions in an image that deviate from their normal appearance. A popular approach trains an auto-encoder on anomaly-free images and performs anomaly detection by calculating the difference between the input and the reconstructed image. This approach assumes that the auto-encoder will be unable to accurately reconstruct anomalous regions. But in practice neural networks generalize well even to anomalies and reconstruct them sufficiently well, thus reducing the detection capabilities. Accurate reconstruction is far less likely if the anomaly pixels were not visible to the auto-encoder. We thus cast anomaly detection as a self-supervised reconstruction-by-inpainting problem. Our approach (RIAD) randomly removes partial image regions and reconstructs the image from partial inpaintings, thus addressing the drawbacks of auto-enocoding methods. RIAD is extensively evaluated on several benchmarks and sets a new state-of-the art on a recent highly challenging anomaly detection benchmark.

Keywords:
Anomaly Detection, Video anomaly detection, Inpainting, CNN

1. Introduction

Anomaly detection focuses on detection of inconsistencies with training data. A common assumption is that the anomalies are rare and diverse, which makes collection and annotation of sufficiently large datasets to train discriminative models unfeasible. Generative approaches are thus preferred. These approaches attempt to capture the distribution of normal data and pinpoint anomalies as outliers. Recent anomaly detection methods thus focus on learning the distribution of anomaly-free data via image reconstruction through an auto-encoder network [1, 2, 3]. Each image is assigned an anomaly score based on the reconstruction error, where the assumption is that the auto-encoder will not be able to reconstruct image patterns, that were not observed during training.

Deep auto-encoder-based anomaly detection approaches learn features that allow reconstructing a variety of objects with a very low reconstruction error. But due to a high generalization capacity of auto-encoders, the anomalies are often reconstructed with a

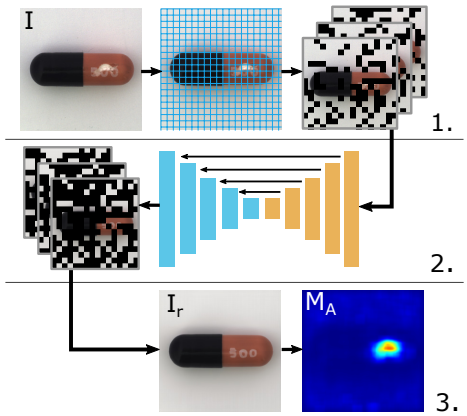

Figure 1: Our anomaly detection method is based on reconstruction-by-inpainting. The image **I** is split into a grid of rectangular regions of the size of $k \times k$ pixels. The set of rectangular regions is randomly split into n disjoint subsets. For each subset, the regions belonging to that subset are removed from the original image (they are set to 0), resulting in n input images (1). The input images are reconstructed by an inpainting network generating n output images, each reconstructing the regions removed in their corresponding input image (2). The individual reconstructed regions from the n partial reconstructions are re-assembled into a single reconstructed image $\mathbf{I}_r$. The reconstruction quality is then evaluated to generate an anomaly map M_A (3).

Email addresses: vitjan.zavrtanik@fri.uni-lj.si (Vitjan Zavrtanik), matej.kristan@fri.uni-lj.si (Matej Kristan), danijel.skocaj@fri.uni-lj.si (Danijel Skoaj)

high fidelity [4, 5]. This violates the core assumption and makes anomalous regions indistinguishable from anomaly-free regions based on the reconstruction error alone.

Self-supervised learning approaches [6, 7] based on geometric transformation pretext tasks such as translation and rotation prediction [8] have recently been proposed for unsupervised feature learning on anomaly-free images. Assuming that the geometric transformation accuracy will drop on anomalous images, the anomaly presence can be detected implicitly from the quality of transformation prediction. These methods provide excellent results on a variety of images, however they perform poorly on geometric transformation invariant images such as rotation invariant textures or symmetrical objects.

We propose a novel anomaly detection method that does not suffer from the over-accurate anomaly reconstruction observed in auto-encoders nor from object geometric transformation invariance observed in the pretext-task-based methods.

Our main contribution lies in casting anomaly detection as a reconstruction-by-inpainting problem (see Figure 1). In contrast to auto-encoders, local regions are reconstructed by conditioning *only* on their immediate neighborhood, excluding the input pixels in the region, which is being reconstructed. The likelihood of accurately reconstructing the anomaly by generalizing the neighborhood appearance is therefore very low. On the other hand, the reconstruction of removed non-anomalous regions is not hampered, since the network is trained on anomaly-free images, and such regions are therefore modelled very well. Consequently, the difference of the reconstruction errors in anomalous and non-anomalous regions increases, which improves the downstream anomaly detection. As a secondary contribution, we propose a gradient similarity [9] based loss function for training as well as an anomaly score estimation function. Our approach sets a new state-of-the-art on a recent challenging anomaly detection benchmark.

The remainder of the paper is structured as follows. In Section 2 we discuss recent approaches for image and video anomaly detection while in Section 3 our reconstruction-by-inpainting anomaly detection method is presented in detail. Section 4 describes the evaluation methodology used, which is followed by presentation of results of our method on the tasks of image anomaly detection and localization and video anomaly detection in Section 5. Additionally, in Section 6 we present the ablation study and conclude the paper with Section 7.

2. Related Work

In many recent anomaly detection methods generative adversarial networks (GAN) [10] are used to learn image representations upon which the anomalies are detected [11, 12]. The networks are commonly adversarially trained auto-encoders trained to reconstruct the input image [1, 2]. Determining whether an image is an anomaly is often based on the reconstruction error. GANomaly [1] trains an adversarial auto-encoder to learn image reconstruction. The anomaly score of an image is defined as the difference between latent space representations of the original image and of the reconstructed image. In [2] the method from [1] is extended by adding skip connections to the adversarial auto-encoder and by changing the anomaly score computation to be based on the reconstruction error in addition to the difference between the latent space representations of the original image and the reconstructed image. Auto-encoder based reconstruction methods are also used in video anomaly detection, where in addition to image reconstruction, motion information is also utilized [13, 14]. Ano-GAN [11] trains a DCGAN [15] to model the input image distribution. During testing an approximate input image latent space representation is found and the input image is reconstructed from this approximation. The anomaly score is calculated from the reconstruction error and from the difference in the discriminator output for the generated and the original image. In [16] an ensamble of student networks are trained to mimic the output of a pretrained teacher method for a given image patch. Similarly to the reconstruction based methods, the anomaly score is based on how well the student networks reconstruct the features extracted by the teacher network.

One-class learning methods can also be considered anomaly detection methods. Deep-SVDD [17] trains the neural network to map the image representations to a minimal volume hypersphere. Out-of-distribution detection (OOD) is then performed by assigning an anomaly score to each image based on the distance to the hypersphere center. Deep-SVDD assumes the data can be accurately represented by a single mode, however this assumption does not always hold.

Many recent self-supervised learning techniques are formulated as pretext tasks, which can be any task that a neural network can learn how to solve and for which explicit labels are not required. Examples of these include transformation predictions [8, 18], cross-channel auto encoding [19, 20] or clustering [21, 22, 23]. Self-supervised learning can also be formulated as a contrastive learning problem [24, 25, 26]. Self-supervised

geometric transformation prediction methods have been applied to the task of anomaly detection [6, 7], but are not robust to objects invariant to the used geometric transformations. Similarly colorization [20] and image rotation have also been applied to anomaly detection in [4].

3. Reconstruction-by-inpainting anomaly detection

Our method is based on an encoder-decoder network trained for image inpainting on anomaly-free samples. First a portion of the input image pixels are removed and the trained network is used to replace the missing information with semantically plausible content. Each image is assigned an anomaly score according to the region with the poorest reconstruction quality. The main steps of our reconstruction-by-inpainting anomaly detection (RIAD) method are shown in Figure 1. In the following, our method is described in detail.

3.1. Reconstruction-by-inpainting formulation

In our approach, randomly selected regions are set to zero in the input image and inpainted by the trained network. In particular, several connected regions are sampled and removed from the image. The input image is altered by removing a set of pixels by first partitioning the input image into square regions of size k. Each image is separated into a grid of dimensions $\frac{H}{k} \times \frac{W}{k}$, where W and H are the width and height of the image. Each grid element is therefore a square of $k \times k$ pixels. The images are resized so that their dimensions are divisible by k. The grid is randomly partitioned into n disjoint sets S_i each containing $\frac{N}{n}$ grid elements, where $N = \frac{H}{k} \times \frac{W}{k}$ is the number of all grid elements. A given set S_i contains approximately $\frac{1}{n}$ of all pixels. The size and portion of the removed regions can be controlled by setting the k and n hyperparameters respectively. A mask $\mathbf{M}_{S_i}$ of the same width and height as the input image $\mathbf{I}$ is generated for each set of regions S_i. $\mathbf{M}_{S_i}$ is a binary mask that contains zeros in regions belonging to S_i. During inference $\mathbf{M}_{S_i}$ is used to set the regions belonging to S_i to zero in image $\mathbf{I}$. Regions set to zero are then reconstructed using the trained network. Examples of network input generation by image masking at several scales k are shown in Figure 2. The entire image is reconstructed by partially reconstructing each set S_i, where $i \in \{1, 2, ..., n\}$, as the union of the regions belonging to sets S_i for all $i \in \{1, 2, ..., n\}$ cover the entire image:

$$\sum_{i=1}^{n} \overline{\mathbf{M}}_{S_i} = \mathbf{1}_{H \times W}, \tag{1}$$

where $\overline{\mathbf{M}}_{S_i}$ is the binary inverse of $\mathbf{M}_{S_i}$ and $\mathbf{1}_{H \times W}$ is a matrix of ones of size $H \times W$.

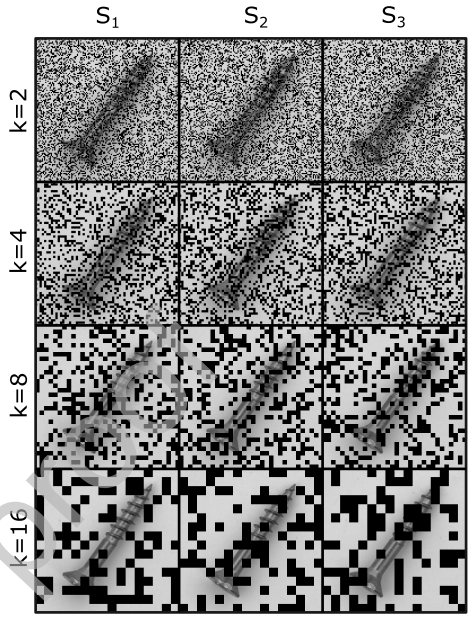

Figure 2: Examples of an input image masked by $n = 3$ disjoint sets of inpainting regions with masked region sizes $k \in \{2, 4, 8, 16\}$. Note that in every row of images, which show masked images using different masked region sizes, each pixel in the image is masked exactly once.

The network takes masked images $\mathbf{I}_i = \mathbf{M}_{S_i} \odot \mathbf{I}$, as shown in Figure 2, as input. The images $\mathbf{I}_i$ are fed into the network sequentially and the network inpaints each image individually. The reconstructed images $\mathbf{I}_{ri}$ are masked and summed into the final reconstruction $\mathbf{I}_r$ (Figure 1) i.e.,

$$\mathbf{I}_r = \sum_{i=1}^{n} \overline{\mathbf{M}}_{S_i} \odot \mathbf{I}_{ri}. \tag{2}$$

The final reconstructed image $\mathbf{I}_r$ is constructed from partially reconstructed images $\mathbf{I}_{ri}$. In each partially reconstructed image $\mathbf{I}_{ri}$, the regions not belonging to S_i are set to zero by multiplying $\mathbf{I}_{ri}$ by $\overline{\mathbf{M}}_{S_i}$. Each $\mathbf{I}_{ri}$ therefore contributes only the regions belonging to S_i that were removed at input. An example of the masking procedure is shown in Figure 3.

3.2. Reconstruction network architecture

A U-Net [27] based encoder-decoder network is used to reconstruct the removed regions. The architecture of the network used is shown in Figure 4. Skip connections are used to transfer features through different layers of

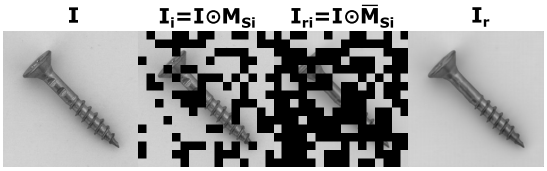

Figure 3: Regions belonging to S_i are set to zero in image $\mathbf{I}$ by multiplying $\mathbf{I}$ by the region mask $\mathbf{M}_{S_i}$. The resulting image $\mathbf{I}_i$ is fed into the network to create a partial reconstruction $\mathbf{I}_{ri}$. The regions in $\mathbf{I}_{ri}$ not belonging to S_i are set to zero by multiplying with the inverse of $\mathbf{M}_{S_i}$. The final image reconstruction $\mathbf{I}_r$ is assembled from partial reconstructions $\mathbf{I}_{ri}$.

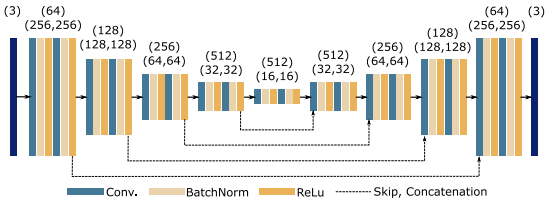

Figure 4: The architecture of the inpainting convolutional neural network used in our method.

the network which leads to an accurate reconstruction of details, that are otherwise difficult to reconstruct. Skip-connections used in a network trained for auto-encoding lead to trivial solutions as low level features are propagated directly to the final layers of the network. Since our method is not conditioned on the pixels being reconstructed, such trivial solutions are avoided.

For training auto-encoders a per-pixel L_2 loss is commonly used, however this assumes independence between neighboring pixels, which is often incorrect. For this reason, losses that penalize structural differences between the reconstructed regions and the regions belonging to the original image are used. Specifically, a multi-scale gradient magnitude similarity (MSGMS) loss, based on [9], is proposed and the structured similarity index (SSIM) [28] loss is used. Both SSIM and MSGMS are utilized as they are patch similarity metrics that focus on different image properties.

The gradient magnitude similarity (GMS) [9] between the original and the reconstructed image is computed by first computing the gradient magnitude maps of the original and the reconstructed image:

$$g(\mathbf{I}) = \sqrt{(\mathbf{I} * \mathbf{h}_x)^2 + (\mathbf{I} * \mathbf{h}_y)^2}, \tag{3}$$

where $g(\mathbf{I})$ is the gradient magnitude map for image $\mathbf{I}$. $\mathbf{h}_x$ and $\mathbf{h}_y$ are 3×3 Prewitt filters along the x and y dimensions and $*$ is the convolution operation. The gradient magnitude similarity map between the original image $\mathbf{I}$ and the reconstructed image $\mathbf{I}_r$ is then defined as:

$$GMS(\mathbf{I}, \mathbf{I}_r) = \frac{2g(\mathbf{I})g(\mathbf{I}_r) + c}{g(\mathbf{I})^2 + g(\mathbf{I}_r)^2 + c}, \tag{4}$$

where c is a constant ensuring numerical stability and helps with mediating the response in noisy areas. We extend GMS [9] to a multi-scale variant (MSGMS) by computing it over several image scales. The MSGMS loss is calculated over an image pyramid of 4 different scales. The smoothed downsampled images are generated at several scales by average pooling each image several times with a sliding windows of size 2×2 and stride of 2. In addition to the original image the resulting image pyramid is composed of images that are $\frac{1}{2}$, $\frac{1}{4}$ and $\frac{1}{8}$ of the original size. The MSGMS loss is defined as the mean value of the GMS distance map at several scales:

$$L_G(\mathbf{I}, \mathbf{I}_r) = \frac{1}{4}\sum_{l=1}^{4}\frac{1}{N_l}\sum_{i=1}^{H_l}\sum_{j=1}^{W_l} 1 - GMS(\mathbf{I}_l, \mathbf{I}_{rl})_{(i,j)}, \tag{5}$$

where H_l and W_l are the height and width of the image and N_l the number of pixels at scale l. The $\mathbf{I}_l$ and $\mathbf{I}_{rl}$ are the original and the reconstruction images at scale l respectively. $GMS(\mathbf{I}_l, \mathbf{I}_{rl})_{(i,j)}$ is the value of the GMS map of $\mathbf{I}_{rl}$ and $\mathbf{I}_l$ at pixel (i, j).

The SSIM loss is defined as:

$$L_S(\mathbf{I}, \mathbf{I}_r) = \frac{1}{N_p}\sum_{i=1}^{H}\sum_{j=1}^{W} 1 - SSIM(\mathbf{I}, \mathbf{I}_r)_{(i,j)}, \tag{6}$$

where $\mathbf{I}$ and $\mathbf{I}_r$ are the original and the reconstructed image, N_p is the number of pixels in the image $\mathbf{I}$, and $SSIM(\mathbf{I}, \mathbf{I}_r)_{(i,j)}$ is the SSIM [28] value between two patches of $\mathbf{I}$ and $\mathbf{I}_r$ centered at (i, j). The total loss takes into account the MSGMS loss and the SSIM loss as well as the pixel-wise L_2 loss for regularization:

$$L = \lambda_G L_G + \lambda_S L_S + L_2, \tag{7}$$

where λ_G and λ_S are the individual loss weights.

3.3. Multi-scale training

The accuracy of region reconstruction depends on the size of the area that has been removed during inference. Since the anomaly detection relies on the reconstruction being as faithful as possible in non-anomalous regions, anomaly detection performance may also depend on the region size k used and on the size of the anomaly that is being reconstructed. If k is much larger than the anomaly, accurate reconstruction through inpainting is

not possible, if k is too small, the inpainting network could infer parts of the anomaly from its surroundings. Since anomalies occur in various sizes, their detection has to consider several scales. A more reliable reconstruction error map can be generated by taking into account multiple reconstructions of an individual image, generated using several k values.

To increase the robustness of anomaly detection, several sizes of masked local regions are considered during training. In particular, the size k is sampled from a set of values $K = \{k_i\}_{i=1:N_K}$, where N_k is the set cardinality. For example, $K = \{2, 4, 8, 16\}$ is used in most of our experiments as it covers a wide range of anomaly scales.

The reconstruction-by-inpainting and the training iteration procedures are written more compactly in Algorithms 1 and 2.

Algorithm 1: Reconstruction by inpainting

input : I ▷ input image
k ▷ region size parameter
output: I_r ▷ reconstructed image
n = number of disjoint sets ▷ hyperparameter
$N = \frac{H}{k} \times \frac{W}{k}$ ▷ number of squared regions
R = permute(N) ▷ randomly permute N indices of regions of $k \times k$ pixels
$S_i = \{R_{i\frac{N}{n}+j}, \text{j} \in \{0, 1, ..., \frac{N}{n}\}\}$ ▷ partition R into
$S = \{S_i \text{ for i} \in \{0,..., \text{n}\}\}$ ▷ n disjoint sets S_i
for S_i *in* S **do**
  $M_{S_i}^{(px)} = \begin{cases} 0, & \text{if } px \in S_i \\ 1, & \text{otherwise} \end{cases}$ ▷ binary mask, where pixels in regions S_i are set to 0
  $I_i = M_{S_i} \odot I$ ▷ mask out part of image
  I_{ri} = `inpainting_model`(I_i) ▷ reconstruct removed regions
end
$I_r = \sum_i^n \overline{M}_{S_i} \odot I_{ri}$ ▷ assemble full image from reconstructed regions of each I_{ri} (2)

Algorithm 2: RIAD training iteration

input : I ▷ input image
output: L ▷ loss for image I
K = set of region size parameters ▷ hyperparam.
k = random sample from K
I_r = `reconstruction_by_inpainting`(I, k)
▷ reconstruct image using Algorithm 1
$L = \lambda_G L_G(I, I_r) + \lambda_S L_S(I, I_r) + L_2(I, I_r)$ ▷ loss (7)

Once the model is trained, it is ready to be used for detection of anomalies.

3.4. Anomaly Detection

The input image $\mathbf{I}$ is first partitioned into a grid which is divided into n disjoint sets S_i. The process for generating the reconstructed image $\mathbf{I}_r$ is the same as during training and is described by (2). The anomaly scores on the pixel as well as on the image level are then calculated by comparing $\mathbf{I}_r$ and $\mathbf{I}$.

The anomaly score map is obtained by first calculating the GMS map considering the input and the reconstructed image over multiple scales. In particular, for each scale l a scaled GMS map is computed by (4) from the input $\mathbf{I}_l$ and reconstructed image $\mathbf{I}_{rl}$ down-sampled to the scale l, using the same down-sampling procedure as when computing the MSGMS loss during training. $GMS(\mathbf{I}_l, \mathbf{I}_{rl})$ is then upsampled to the original resolution. A multiscale GMS map $MSGMS(\mathbf{I}, \mathbf{I}_r)$ is then computed as the per-pixel average of the scaled GMS maps. Anomalies tend to occupy larger spatially connected regions, therefore the reconstruction error can be aggregated over a larger region for a more accurate anomaly detection. The MSGMS map is thus post-processed by a mean-filter convolution and subtracted from a matrix of ones to generate the anomaly map $G(\mathbf{I}, \mathbf{I}_r) \in [0, 1]^{H \times W}$:

$$G(\mathbf{I}, \mathbf{I}_r) = \mathbf{1}_{H \times W} - (MSGMS(\mathbf{I}, \mathbf{I}_r) * \mathbf{f}_{s_f \times s_f}) \ , \quad (8)$$

where $\mathbf{f}_{s_f \times s_f}$ is the mean filter of size $(s_f \times s_f)$ used for smoothing, $*$ is the convolution operation, and $\mathbf{1}_{H \times W}$ is a matrix of ones of size $H \times W$. $MSGMS(\mathbf{I}, \mathbf{I}_r)$ is the MSGMS map generated from $\mathbf{I}$ and $\mathbf{I}_r$. Smoothing improves the detection robustness in presence of high MSGMS values in small regions that result from poor reconstruction of non anomalous objects or background noise rather than actual anomalies.

During inference an individual image is masked and inpainted several times for each $k \in K$. The output of the method is then constructed as an average of the anomaly maps G generated for each image reconstruction at each k:

$$G_A(\mathbf{I}, \mathbf{I}_r) = \frac{1}{N_K} \sum_{k \in K} G(\mathbf{I}, \mathbf{I}_r)_k \ , \quad (9)$$

where $G(\mathbf{I}, \mathbf{I}_r)_k$ is the anomaly map generated as defined in (8), using the anomaly size masking value k.

Finally, the image level anomaly score $\epsilon(\mathbf{I}, \mathbf{I}_r)$ is computed by taking the maximum of $G_A(\mathbf{I}, \mathbf{I}_r)$, i.e.,

$$\epsilon(\mathbf{I}, \mathbf{I}_r) = max(G_A(\mathbf{I}, \mathbf{I}_r)) \ , \quad (10)$$

where $\mathbf{I}$ is the input image and $\mathbf{I}_r$ is the final reconstructed image, indicating whether an image contains

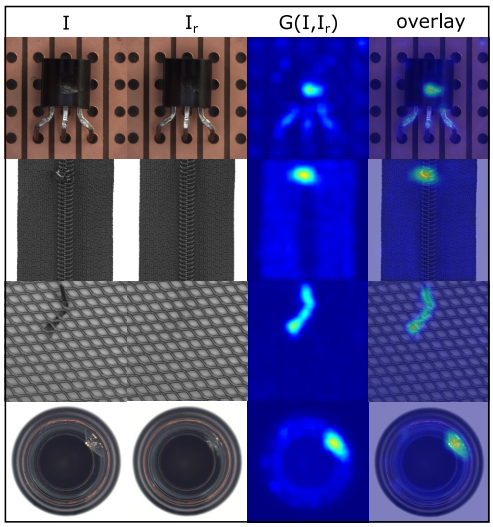

Figure 5: Reconstruction and anomaly score estimation examples of our method. $\mathbf{I}$ is the input image, $\mathbf{I}_r$ is the fully reconstructed image and $G(\mathbf{I}, \mathbf{I}_r)$ is the MSGMS based anomaly map. The overlay of the anomaly map over the original image visualizes the localization performance of RIAD.

an anomaly or not. Examples of anomalous images $\mathbf{I}$, their reconstructions $\mathbf{I}_r$ and locally smoothed anomaly maps $G(\mathbf{I}, \mathbf{I}_r)$ are shown in Figure 5. The RIAD inference process is written more compactly in Algorithm 3.

Algorithm 3: RIAD inference

input : I ▷ input image
output: ϵ ▷ anomaly score (for detection)
G_A ▷ anomaly map (for localisation)
K = set of region size parameters ▷ hyperparam.
for k *in* K **do**
  I_r=reconstruction_by_inpainting(I, k)
  ▷ reconstruct image using Algorithm 1
  $G(I, I_r)_k = 1_{H \times W} - (MSGMS(I, I_r) \times f_{sf \times sf})$
  ▷ anomaly map for region size k (8)
end
$G_A(I, I_r) = \frac{1}{N_k} \sum_{k \in K} G(I, I_r)_k$ ▷ anomaly map (9)
$\epsilon(I, I_r) = max(G_A(I, I_r))$ ▷ anomaly score (10)

4. Experiments

Our reconstruction-by-inpainting based anomaly detection (RIAD) is evaluated on the challenging MVTec [29] dataset on the tasks of anomaly detection and localization. RIAD is compared with current state-of-the-art methods. Additionally, the generality of RIAD is demonstrated by evaluating our method on the UCSD Ped2 [30] and Avenue [31] video anomaly detection datasets. The main experimental results are presented in Section 5, while the ablation study is reported in Section 6.

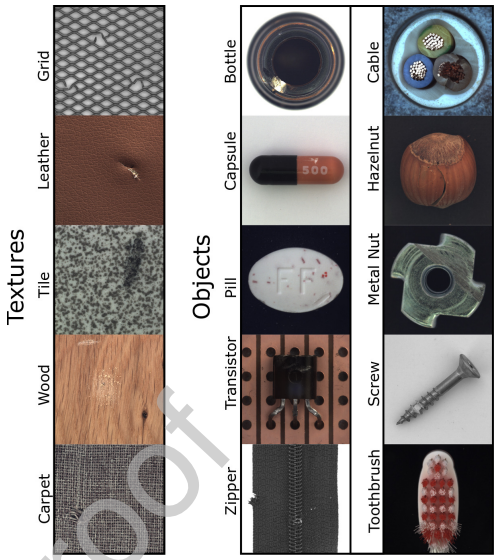

Figure 6: Examples of anomalous images for each class of the MVTec anomaly dataset. The classes are split into objects and textures.

4.1. Datasets

The MVTec anomaly detection dataset [29] is the most recent challenging anomaly dataset containing a variety of faulty products taken in a controlled environment and constitutes a realistic anomaly detection problem. The dataset contains 3629 training images of non anomalous objects and 1725 test images containing various types of anomalies as well as non-anomalous samples. The dataset is made up of 15 different object classes with images for each class split into a training and a testing set. It includes a detailed ground truth with pixel-wise mask annotations for each anomalous region. The image classes are further grouped as textures and objects. Examples of anomalous images are shown in Figure 6.

The generality of RIAD is further demonstrated on the UCSD Ped2 [30] and Avenue[31] video anomaly detection datasets. The Ped2 dataset contains 16 training videos and 12 testing videos from a surveillance camera with a static background. The footage contains various groups of people walking parallel to the camera. Unusual objects in the scene such as bicycles and cars are marked as anomalous. The Avenue dataset has 30652

total frames split into 16 training and 21 testing videos. The footage was captured by a surveillance camera and has a static background. Anomalies are unusual activities such as walking too close to the camera, running and loitering. In several videos unusual static objects are present that do not appear in the training data but are not marked as anomalies. Due to the ambiguity of these objects Hinami et. al. [32] propose evaluating on the Avenue17 dataset which does not contain the videos containing these static objects. We therefore evaluate the proposed method on Avenue17.

4.2. Evaluation protocol

The proposed method is evaluated on two tasks: image-level anomaly detection and pixel-level anomaly localisation. The standard ROC AUC is used as the primary metric as it is the standard evaluation metric in both image and video anomaly detection works [33, 1, 2, 14, 13, 11]. Additionally, a standard pixel based ROC AUC score is used as an anomaly localization metric. We compare the anomaly localization ability of our method with other state-of-the-art anomaly localization methods AE-SSIM [34], AnoGAN [11], Visually Explainable Auto Encoders (VEVAE) [35] and our implementation of the most recent Uninformed Students method [16] on the MVTec[29] data. The network is trained for each object class individually and the class-wise ROC AUC score is reported. We compare RIAD to recently proposed anomaly detection approaches GANomaly [1], GeoTrans [6] and Inverse Transform Auto-Encoders [4].

We reimplement the multi-scale version of the current state-of-the-art Uninformed Students (US) method [16] and evaluate it on the task of anomaly detection and localization. In [16], US method is only evaluated on the task of anomaly localization and the authors provide no procedure for producing image level results. The output of US is an anomaly score map which can be condensed into a single anomaly score for each image using the same interpretation of the anomaly score map as is used in our method. The anomaly score map of our reimplementation of US is therefore interpreted by taking its maximum to generate the anomaly score as described in (10). This allows us to evaluate the Uninformed Students method as an anomaly detection method and provide a fair comparison to other approaches.

4.3. Implementation details

A U-Net [27] based network architecture is used and is shown in Figure 4. All experiments on the MVTec dataset are run for 300 epochs with a batch size of 8. The network is trained using the Adam optimizer with a learning rate of 0.0001 which is reduced by a factor of 10 after 250 epochs. λ_S and λ_G are set to 1. In a practical application, the weights of individual losses in (7) could be tuned on a validation set. However, in our setup the validation sets are unavailable and we set all weights to an equal value. Unless stated otherwise, the method used in the experiments uses the parameters $n = 3$ and varying region size ensambling with $K = \{2, 4, 8, 16\}$. While the values in K are application specific and can be tuned to a specific anomaly detection task, we include a wide range of scales into our approach for generality of the method. Values larger than $k = 16$ are not used in our experiments as we have found that they do not significantly improve or degrade the performance, but adding them into K increases the overall complexity. The mean-filter of size (21×21) is used for smoothing the MSGMS maps in all experiments.

Due to the smaller scale of the objects, $K = \{2, 4\}$ is used for region size ensambling on the Ped2 dataset and $K = \{2, 4, 8, 16\}$ is used for the Avenue17 dataset, as with experiments performed on the MVTec dataset. To account for temporal smoothness, the anomaly score is averaged over the past three frames on the Ped2 and Avenue17 datasets.

5. Results

Anomaly Detection Anomaly detection results on the MVTec data are shown in Table 1. RIAD achieves the highest average ROC-AUC, outperforming the best state-of-the-art method by 4 percentage points and on texture classes by 3.6 percentage points. US [16] performs better on object classes such as cable and hazelnut that contain large regions with a significant amount of noise and are therefore hard to inpaint. However, RIAD outperforms US on 9 classes and performs very well on structured pattern textures and objects such as bottle, grid and zipper, where little random pattern regions are present and where non anomalous regions can be accurately inferred from its surroundings, thus reducing the possibility of a false positive detection. Figure 7 contains examples of anomaly maps produced by RIAD for MVTec images. For structured object classes such as grid or bottle, RIAD produces anomaly maps that clearly localize the anomalies. In RIAD anomaly maps for random-pattern-heavy classes such as tile or metal nut, anomalous regions are still assigned high anomaly scores and can be separated from non-anomalous regions. However, the random pattern regions result in increased reconstruction error, which results in increased anomaly score in these non-anomalous regions.

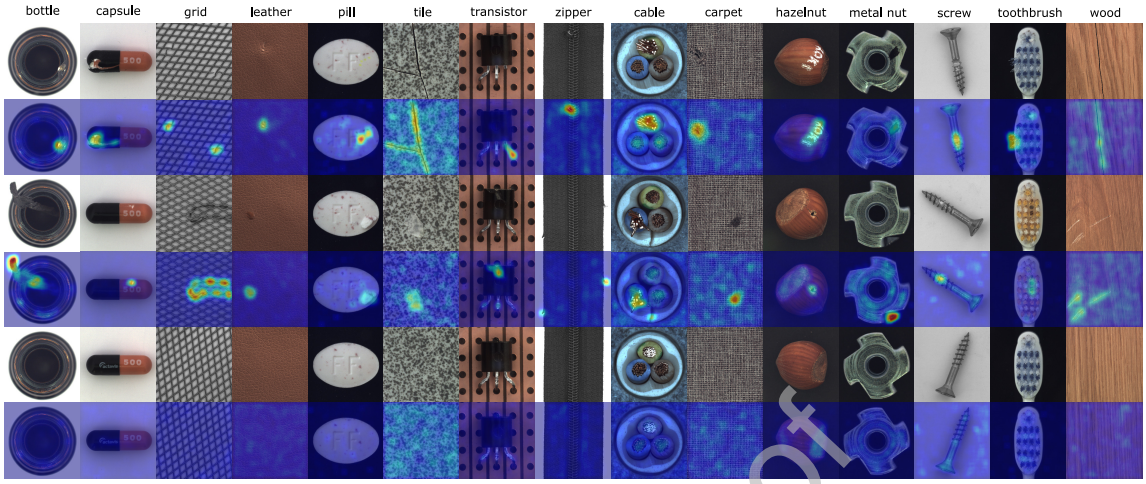

Figure 7: Qualitative results of RIAD on the MVTec data containing anomalous images (row 1 and 3) and overlaid anomaly maps produced by RIAD (row 2 and 4). Row 5 contains non-anomalous images and row 6 contains the corresponding anomaly maps.

Anomaly Localization The results for anomaly localization on the MVTec data are listed in Table 2. RIAD achieves a state-of-the-art overall ROC-AUC score for anomaly localization, outperforming all of the tested state-of-the-art methods. RIAD outperforms the best recent state-of-the-art method US [16] on 8 classes and achieves a higher overall ROC-AUC score for anomaly localization. Additionally, RIAD is simpler than US, as it trains only a single model, compared to the multi-scale version of US, which requires an ensemble of 9 separately trained models.

Video Anomaly Detection To demonstrate its effectiveness, we evaluate RIAD on the task of video anomaly detection and compare it to recent state-of-the-art approaches. Recent video anomaly detection methods heavily rely on motion information. As RIAD was not particularly designed for video anomaly detection it does not use motion information and is therefore evaluated under an unfavourable setup compared to state-of-the-art approaches. Despite relying on individual frames, however, RIAD achieves favourable results on video anomaly detection datasets due to strong image-based anomaly detection capability. Results in Table 3 show that RIAD performs comparably to some state-of-the-art methods such as [32] and [36] that utilize motion information, indicating the potential to further boost the performance of video anomaly detection methods by using the concepts introduced in RIAD.

Figure 8 shows examples of anomalies and their reconstructions by RIAD. Observe that the network cannot reconstruct the anomalies since these were not ob-

Class	GeoTrans[6]	GANomaly[1]	ITAE[4]	US[16]	RIAD
bottle	74.4	89.2	94.1	99.0	**99.9**
capsule	67.0	73.2	68.1	86.1	**88.4**
grid	61.9	70.8	88.3	81.0	**99.6**
leather	84.1	84.2	86.2	88.2	**100**
pill	63.0	74.3	78.6	**87.9**	83.8
tile	41.7	79.4	73.5	**99.1**	98.7
transistor	86.9	79.2	84.3	81.8	**90.9**
zipper	82.0	74.5	87.6	91.9	**98.1**
cable	78.3	75.7	83.2	**86.2**	81.9
carpet	43.7	69.9	70.6	**91.6**	84.2
hazelnut	35.9	78.5	85.5	**93.1**	83.3
metal nut	81.3	70.0	66.7	82.0	**88.5**
screw	50.0	74.6	**100**	54.9	84.5
toothbrush	97.2	65.3	**100**	95.3	**100**
wood	61.1	83.4	92.3	**97.7**	93.0
avg_{tex}	58.5	76.5	82.2	91.5	**95.1**
avg_{obj}	71.6	75.4	84.8	85.8	**89.9**
avg	67.2	76.2	83.9	87.7	**91.7**

Table 1: Results for the task of anomaly detection on the MVTec dataset. Results are listed as ROC AUC scores and are marked individually for each class. An average score over all classes is also reported in the last row and the average score over the texture classes is computed in the row marked avg_{tex} and over the object classes in the row marked avg_{obj}.

Class	AE-SSIM[34]	AnoGAN[11]	VEVAE[35]	US[16]	RIAD
bottle	93.0	86.0	87.0	97.8	**98.4**
capsule	94.0	84.0	74.0	**96.8**	92.8
grid	94.0	58.0	73.0	89.9	**98.8**
leather	78.0	64.0	95.0	97.8	**99.4**
pill	91.0	87.0	83.0	**96.5**	95.7
tile	59.0	50.0	80.0	**92.5**	89.1
transistor	80.0	80.0	**93.0**	73.7	87.7
zipper	88.0	78.0	78.0	95.6	**97.8**
cable	82.0	78.0	90.0	**91.9**	84.2
carpet	87.0	54.0	78.0	93.5	**96.3**
hazelnut	97.0	87.0	98.0	**98.2**	96.1
metal nut	89.0	76.0	94.0	**97.2**	92.5
screw	96.0	80.0	97.0	97.4	**98.8**
toothbrush	92.0	90.0	94.0	97.9	**98.9**
wood	73.0	62.0	77.0	**92.1**	85.8
avg_{tex}	78.2	57.7	80.6	93.2	**93.9**
avg_{obj}	90.2	82.6	88.8	**94.3**	**94.3**
avg	86.2	74.3	86.1	93.9	**94.2**

Table 2: Localization evaluated by ROC AUC scores on the MVTec dataset. Our method ourperforms the SSIM-AE, AnoGAN, VEVAE and US methods in the overall anomaly score map ROC AUC metric.

Method	Ped2	Avenue17
RIAD	92.5	88.9
Appearance Motion AE [13]	96.2	-
Future Frame Prediction [14]	95.4	-
Object Centric AE [33]	97.8	91.6
Growing Gas [37]	94.1	-
FRCN-action [32]	92.2	89.8
Conv-AE [36]	90.0	76.9

Table 3: Video anomaly detection results compared to state-of-the-art methods.

served during training and were partially covered during inference.

6. Ablation Studies

Region size and number of subsets The results for varying region size parameter k are listed in Table 4. To evaluate the impact of region size k, we remove the varying region size ensambling component of our method and train multiple models using a single k. A single k value is also used during evaluation. The region size parameter can have a significant impact on the result of individual classes. A low $k \in \{2, 4\}$ setting performs well in texture classes, while a high $k \in \{8, 16\}$ performs better in object classes where the method has to learn to reconstruct larger object parts. The reconstruction of anomalies is difficult regardless of the k parameter setting since the network has to infer the anomalous regions from the surrounding area which is still much more demanding than merely reconstructing the anomalies using an auto-encoder. Examples of anomaly

Figure 8: Anomalies in the Ped2 data set. (Left) Original images containing anomalies. (Right) Frames reconstructed by our method. Anomalies and their reconstructions are marked in green.

maps generated by networks trained at varying region size parameters are shown in Figure 9.

The results for varying the portion of total pixels removed are shown in Table 5. While the percentage of the removed information does impact the performance on individual classes, we observe a stable performance over a range of n, dropping only at $n = 2$, where half of the image information is removed. The results thus show considerable robustness of RIAD to the portion of image information removed.

Skip-connections Skip-connections enable the propagation of features at various scales throughout the network. We examine their effect by training the encoder-decoder network in RIAD with skip connections removed ($RIAD_{ns}$). The results are listed in Table 6. The usefulness of skip connections depends on the type of the examined object. While they increase performance in texture classes where low level structures are important for accurate reconstruction, the performance decreases when skip connections are used for reconstructing regions containing significant amount of noise. Low level features describing the noisy textures get propagated through the skip connections during training. This enables the network to better reconstruct detailed textures during training, but also causes higher anomaly score in regions that are difficult to inpaint accurately due to their lack of correlation to the surrounding regions.

Anomaly score function Since the network was trained to minimize both $MSGMS$ and $SSIM$ functions, both can be used as anomaly score estimation functions. The results for the same network using ei-

Class	2	4	8	16
bottle	99.8	99.7	**99.9**	98.8
capsule	84.2	**96.3**	94.6	**96.3**
grid	99.0	98.5	99.5	**99.7**
leather	99.0	99.9	**100**	**100**
pill	79.2	**86.2**	78.5	67.2
tile	**98.9**	92.8	75.8	65.1
transistor	83.2	84.0	91.3	**91.8**
zipper	**99.5**	98.8	97.4	98.1
cable	55.7	60.8	74.4	**87.4**
carpet	82.0	**83.5**	73.8	66.0
hazelnut	64.1	76.6	**91.0**	88.4
metal nut	72.5	62.5	**88.2**	86.3
screw	83.7	**91.1**	86.0	85.1
toothbrush	99.1	99.7	99.8	**99.9**
wood	90.9	**92.1**	86.4	84.4
avg	86.1	88.2	**89.1**	87.6

Table 4: ROC-AUC anomaly detection results of our method trained and evaluated on a single value of k, where $k \in \{2, 4, 8, 16\}$. The top row shows the region size k at which the experiments were ran.

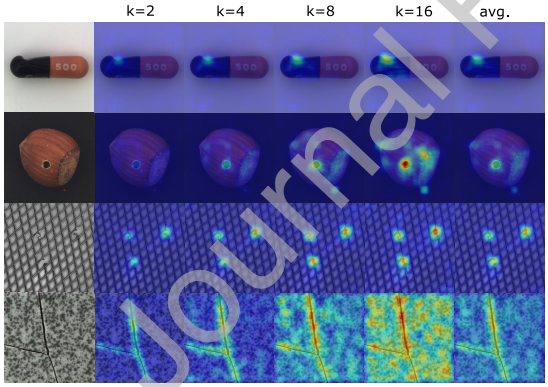

Figure 9: Anomaly maps produced by RIAD with a single $k \in \{2, 4, 8, 16\}$. With rising region size k, the anomalies become harder to reconstruct resulting in higher anomaly scores in anomalous regions, however non-anomalous regions become harder to reconstruct as well. We can see that while anomalous regions are more pronounced in the anomaly maps at higher k values, non-anomalous regions can also be assigned higher anomaly scores. This is especially apparent in random-pattern heavy regions. Using varying region ensambling (column 6) helps with generating high anomaly score values at anomalous regions while maintaining a low score on non-anomalous regions.

Class	n=2	n=3	n=4	n=5
bottle	98.9	99.9	99.8	99.5
capsule	96.0	94.6	96.9	97.1
grid	99.5	99.5	99.3	99.5
leather	99.9	100	100	99.9
pill	71.3	78.5	77.0	77.9
tile	67.8	75.8	71.5	78.0
transistor	88.1	91.3	91.4	85.6
zipper	96.9	97.4	98.2	97.6
cable	87.4	74.4	71.5	67.5
carpet	62.7	73.8	79.0	82.7
hazelnut	73.2	91.0	86.2	85.8
metal nut	80.5	88.2	86.6	86.7
screw	87.5	86.0	89.5	90.3
toothbrush	99.9	99.8	100	100
wood	83.9	86.4	86.0	83.3
avg	86.2	89.1	88.8	88.7

Table 5: Anomaly detection results of RIAD using various portions of masked pixels controlled by the value of $n \in \{2, 3, 4, 5\}$. The box size parameter is fixed and is $k = 8$. Results are listed as ROC-AUC scores.

ther $SSIM$ or $MSGMS$ as anomaly score estimation functions can be seen in Table 6. We can see an improvement of 6.5 percentage points in the overall ROC AUC when using $MSGMS$, due to its higher robustness to random pattern regions which significantly improves the results on hazelnut and metal nut in which random patterns cover the majority of the image regions. $SSIM$ performs better on objects pill and screw, where the visual properties of anomalies are either color based and thus may not generate significant gradient magnitude errors when reconstructed or the images contain non anomalous regions which contain high gradient magnitude values and are difficult to reconstruct once masked, causing high $MSGMS$ errors.

The ablation study shows that different settings of the individual components of the proposed method influence the overall performance. While the performance of the selected components and parameters achieves on average the best results, we can see that the results could be even further improved by prior knowledge about the individual objects and possible defects, such as the size of potential defects and the amount of random-pattern regions present on the object.

7. Conclusion

We address the challenging problem of visual anomaly detection. We proposed a novel reconstruction-by-inpainting based visual anomaly detection method (RIAD) and evaluate it on realistic MVTec anomaly detection dataset.

Class	$RIAD_{ns}$	$RIAD_{SSIM}$	$RIAD$
bottle	99.3	98.8	99.9
capsule	76.2	76.5	88.4
grid	98.5	99.7	99.6
leather	99.9	91.9	100
pill	84.6	93.2	83.8
tile	84.4	99.1	98.7
transistor	99.4	86.6	90.9
zipper	94.3	99.7	98.1
cable	89.3	59.3	81.9
carpet	62.9	86.1	84.2
hazelnut	93.1	62.6	83.3
metal nut	89.0	44.2	88.5
screw	64.1	93.9	84.5
toothbrush	99.1	96.1	100
wood	94.3	90.1	93.0
avg	88.6	85.2	91.7

Table 6: Anomaly detection results of our method with ($RIAD$) and without skip connections ($RIAD_{ns}$). $RIAD_{SSIM}$ uses $SSIM$ as the anomaly score estimation method, while $RIAD_{ns}$ and $RIAD$ use the $MSGMS$ for anomaly score estimation. Results are listed as ROC-AUC scores.

RIAD achieves the state-of-the-art results for image anomaly detection, outperforming the best state-of-the-art by 4 percentage points. RIAD also achieves state-of-the-art results for the task of anomaly localization.

Gradient similarity based method MSGMS was proposed for training and anomaly score estimation. We observed that MSGMS outperforms SSIM as an anomaly score, which makes it potentially useful also for auto-encoder-based methods.

The generality of RIAD was demonstrated on video anomaly detection and evaluated on the Ped2 and Avenue17 datasets. Even though RIAD does not consider a sequence of images, it achieves favourable performance on par with far more complex video anomaly state-of-the-art. The performance could likely further be improved by adding a motion component such as future frame prediction to the training and inference. These will be considerations of our future extensions.

8. Acknowledgement

This work was supported by the Slovenian Research Agency (ARRS) projects J2-9433, J2-8175 and program P2-0214.

References

[1] S. Akcay, A. Atapour-Abarghouei, T. P. Breckon, Ganomaly: Semi-supervised anomaly detection via adversarial training, in: Asian Conference on Computer Vision, Springer, 2018, pp. 622–637.

[2] S. Akçay, A. Atapour-Abarghouei, T. P. Breckon, Skip-ganomaly: Skip connected and adversarially trained encoder-decoder anomaly detection, arXiv preprint arXiv:1901.08954.

[3] H. Zenati, M. Romain, C.-S. Foo, B. Lecouat, V. Chandrasekhar, Adversarially learned anomaly detection, in: 2018 IEEE International Conference on Data Mining (ICDM), IEEE, 2018, pp. 727–736.

[4] C. Huang, J. Cao, F. Ye, M. Li, Y. Zhang, C. Lu, Inverse-transform autoencoder for anomaly detection, arXiv preprint arXiv:1911.10676.

[5] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, A. v. d. Hengel, Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 1705–1714.

[6] I. Golan, R. El-Yaniv, Deep anomaly detection using geometric transformations, in: Advances in Neural Information Processing Systems 31, 2018, pp. 9758–9769.

[7] D. Hendrycks, M. Mazeika, S. Kadavath, D. Song, Using self-supervised learning can improve model robustness and uncertainty, in: Advances in Neural Information Processing Systems, 2019, pp. 15637–15648.

[8] S. Gidaris, P. Singh, N. Komodakis, Unsupervised representation learning by predicting image rotations, in: International Conference of Learning Representations, 2018.

[9] W. Xue, L. Zhang, X. Mou, A. C. Bovik, Gradient magnitude similarity deviation: A highly efficient perceptual image quality index, IEEE Transactions on Image Processing 23 (2) (2013) 684–695.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in neural information processing systems, 2014, pp. 2672–2680.

[11] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, G. Langs, Unsupervised anomaly detection with generative adversarial networks to guide marker discovery, in: International Conference on Information Processing in Medical Imaging, Springer, 2017, pp. 146–157.

[12] T. Schlegl, P. Seeböck, S. M. Waldstein, G. Langs, U. Schmidt-Erfurth, f-anogan: Fast unsupervised anomaly detection with generative adversarial networks, Medical image analysis 54 (2019) 30–44.

[13] T.-N. Nguyen, J. Meunier, Anomaly detection in video sequence with appearance-motion correspondence, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 1273–1283.

[14] W. Liu, W. Luo, D. Lian, S. Gao, Future frame prediction for anomaly detection–a new baseline, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6536–6545.

[15] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, arXiv preprint arXiv:1511.06434.

[16] P. Bergmann, M. Fauser, D. Sattlegger, C. Steger, Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings, arXiv preprint arXiv:1911.02357, Accepted at CVPR 2020.

[17] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, M. Kloft, Deep one-class classification, in: Proceedings of the 35th International Conference on Machine Learning, Vol. 80, 2018, pp. 4393–4402.

[18] M. Noroozi, P. Favaro, Unsupervised learning of visual representations by solving jigsaw puzzles, in: European Conference on Computer Vision, Springer, 2016, pp. 69–84.

[19] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, A. A. Efros,

Context encoders: Feature learning by inpainting, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2536–2544.
[20] R. Zhang, P. Isola, A. A. Efros, Colorful image colorization, in: European conference on computer vision, Springer, 2016, pp. 649–666.
[21] M. Caron, P. Bojanowski, A. Joulin, M. Douze, Deep clustering for unsupervised learning of visual features, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 132–149.
[22] M. Caron, P. Bojanowski, J. Mairal, A. Joulin, Unsupervised pre-training of image features on non-curated data, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 2959–2968.
[23] X. Ji, J. F. Henriques, A. Vedaldi, Invariant information clustering for unsupervised image classification and segmentation, ICCV.
[24] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, Momentum contrast for unsupervised visual representation learning, arXiv preprint arXiv:1911.05722, Accepted at CVPR 2020.
[25] O. J. Hénaff, A. Razavi, C. Doersch, S. Eslami, A. v. d. Oord, Data-efficient image recognition with contrastive predictive coding, arXiv preprint arXiv:1905.09272.
[26] A. v. d. Oord, Y. Li, O. Vinyals, Representation learning with contrastive predictive coding, arXiv preprint arXiv:1807.03748.
[27] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical image computing and computer-assisted intervention, Springer, 2015, pp. 234–241.
[28] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE transactions on image processing 13 (4) (2004) 600–612.
[29] P. Bergmann, M. Fauser, D. Sattlegger, C. Steger, Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 9592–9600.
[30] V. Mahadevan, W. Li, V. Bhalodia, N. Vasconcelos, Anomaly detection in crowded scenes, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 1975–1981.
[31] C. Lu, J. Shi, J. Jia, Abnormal event detection at 150 fps in matlab, in: Proceedings of the IEEE international conference on computer vision, 2013, pp. 2720–2727.
[32] R. Hinami, T. Mei, S. Satoh, Joint detection and recounting of abnormal events by learning deep generic knowledge, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 3619–3627.
[33] R. T. Ionescu, F. S. Khan, M.-I. Georgescu, L. Shao, Object-centric auto-encoders and dummy anomalies for abnormal event detection in video, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 7842–7851.
[34] P. Bergmann, S. Löwe, M. Fauser, D. Sattlegger, C. Steger, Improving unsupervised defect segmentation by applying structural similarity to autoencoders, arXiv preprint arXiv:1807.02011.
[35] W. Liu, R. Li, M. Zheng, S. Karanam, Z. Wu, B. Bhanu, R. J. Radke, O. Camps, Towards visually explaining variational autoencoders, arXiv preprint arXiv:1911.07389, Accepted at CVPR 2020.
[36] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, L. S. Davis, Learning temporal regularity in video sequences, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 733–742.
[37] Q. Sun, H. Liu, T. Harada, Online growing neural gas for anomaly detection in changing surveillance scenes, Pattern Recognition 64 (2017) 187–201.

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Vitjan Zavrtanik is a Ph.D. student at the Visual Cognitive Systems Laboratory at the Faculty of Computer and Information Science at the University of Ljubljana. He completed his master's studies at the University of Ljubljana in 2018. His research interests include anomaly detection, self-supervised learning and object detection.

Matej Kristan Received the Ph.D. degree from the Faculty of Electrical Engineering, University of Ljubljana, in 2008. He is an associate professor at the Faculty of Computer and Information Science, University of Ljubljana. His research interests include probabilistic methods for computer vision with focus on visual tracking, semantic segmentation, object detection and online learning, and computer vision for autonomous robots. He is a member of the IEEE.

Danijel Skočaj is an associate professor at the University of Ljubljana, Faculty of Computer and Information Science. He is the head of the Visual Cognitive Systems Laboratory. He obtained the Ph.D. in computer and information science from the University of Ljubljana in 2003. His main research interests lie in the fields of computer vision, machine learning, and cognitive robotics.