HaloAE: An HaloNet based Local Transformer Auto-Encoder for Anomaly Detection and Localization

Emilie Mathian[1,3], Huidong Liu[2], Lynnette Fernandez-Cuesta[1], Dimitris Samaras[2], Matthieu Foll[1], and Liming Chen[3]

[1] International Agency for Research on Cancer (IARC-WHO), Lyon, France
mathiane@iarc.who.int
[2] Stony Brook University, New York, USA
[3] Ecole Centrale de Lyon, Ecully, France
liming.chen@ec-lyon.fr

Abstract. Unsupervised anomaly detection and localization is a crucial task as it is impossible to collect and label all possible anomalies. Many studies have emphasized the importance of integrating local and global information to achieve accurate segmentation of anomalies. To this end, there has been a growing interest in Transformer, which allows modeling long-range content interactions. However, global interactions through self attention are generally too expensive for most image scales. In this study, we introduce HaloAE, the first auto-encoder based on a local 2D version of Transformer with HaloNet. With HaloAE, we have created a hybrid model that combines convolution and local 2D block-wise self-attention layers and jointly performs anomaly detection and segmentation through a single model. We achieved competitive results on the MVTec dataset, suggesting that vision models incorporating Transformer could benefit from a local computation of the self-attention operation, and pave the way for other applications[4].

Keywords: Anomaly detection, HaloNet, Transformer, auto-encoder.

1 Introduction

Anomaly detection (AD) aims to determine whether a given image contains an abnormal pattern, given a set of normal or abnormal images, while its localization or segmentation need further to determine the subregions containing the anomalies (see Fig.1). This is a common problem in various domains, *e.g.*, in industry to detect defective objects [6], [7], [17],

in medicine [43], [44], [45], or for video surveillance [1], [32], [49]. Listing all anomalies is a difficult task because of their low probability density. Therefore, the problem is usually addressed via unsupervised learning approaches,

[4] The code is available at: https://anonymous.4open.science/r/HaloAE-8313/README.md

also called one-class classification or out-of-distribution detection. The models use only the defect-free samples during the learning phase and attempt to identify and localize anomalies at the time of inference.

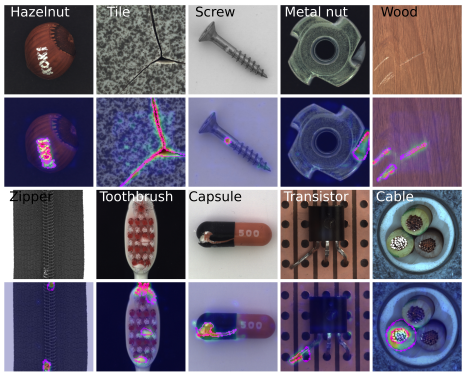

Fig. 1. Anomaly localization results from the MVTec AD dataset. The first and third rows show the input images, the second and fourth rows show the anomaly maps generated by HaloAE, the ground truth localization is circled with a pink line.

State of the art has featured two main approaches on AD: distribution-based or reconstruction. Distribution-based approaches generally make use of Deep Convolutional Neural Networks (Deep CNN) to extract representations of normal images or image patches and learn a parametric distribution, *e.g.*, Gaussian distribution, of these deep features. They typically require to learn two models, one for anomaly detection and another for anomaly segmentation. Reconstruction-based approaches involve training a convolutional auto-encoder [6], [8], [56], [57], or Generative Adversarial Networks (GAN) [4], [44], [42], [2], [3], [54] to reconstruct the normal images assuming that the model should fail to reconstruct abnormal images. The advantage of such approaches is that a single model can be used for both anomaly detection and segmentation. However, most of these models [6], [8], [44], [2] do not perform well as they generalize strongly and can reconstruct anomalies.

Given the fact that detection of abnormal patterns requires combining local and global information, different models have been proposed, either using a fully convolutional neural network (FCNN), [58], or by integrating Transformer's self-attention, which models content long-range interactions [53]. To adapt Transformer to images, Visual Tranformer (ViT) [18] is typically used in an AE [35], [38], [55]. While CNN enables to capture easily translation invariant local patterns, capturing long-range dependencies is challenging because of CNN's poor scaling properties with respect to large receptive fields. On the other hand, Transformer enables to model dependencies between distant elements of a sequence through self-attention but the complexity of memory and computation

grows quadratically with image size. Furthermore, ViT only encodes inter-patch correlations while ignoring intra-patch correlations [46].

In this work, we introduce HaloAE, which extends the block-based local self-attention proposed by HaloNet [52] to AE. Our model allows us to compute intra-patch correlations unlike models using ViT [18]. To regularize the proposed AE and capture a multiscale and semantic rich description of an image, multi-scale feature maps are first generated via a pre-trained network as in [47], [55], [35] and reconstructed. In addition, our model also incorporates a self-supervised learning (SSL) approach to further mitigate the generalization problem of AE. We reused the framework presented by Cut&Paste [31] which defines a proxy classification task between normal and artificially damaged images, which mimic real industrial defects. The performance of HaloAE was evaluated on the challenging MvTec dataset [6], an industrial dataset with 15 object classes (see Fig.1).

The contributions of the paper are threefold:

- We propose HaloAE, a local self-attention-oriented AE based on HaloNet;
- We apply the proposed AE to unsupervised anomaly detection and generate a single model for both anomaly detection and localization;
- We show experimentally that the proposed local self-attention-based AE achieves competitive results on the MVTec benchmark.

2 Related Work

2.1 Anomaly Detection and Localization Models

Reconstruction Based Methods: They are the most commonly used methods for AD and localization [6], [8], [56], [51]. They are usually based on Convolutional AE (CAE), trained to reconstruct defect-free images. At the inference time, the trained models are expected to fail to reconstruct abnormal regions, as they differ from the observed training data. Segmentation maps of abnormal regions are obtained by per-pixel comparison between input and output images based on L_2 deviations [6], [56], or SSIM values [6]. While simple and elegant in design, CAEs suffer from memory and generalize abnormal regions quite well [56], [4]. The latent space regularization enabled by variational auto-encoder (VAE) eases the identification of abnormal samples [51]. Other approaches rely on the improvement of the quality of the reconstructed images using GANs [4], [44],[2], [3], [54] .

It has been shown that while GANs produce sharp images, they are unstable and tend to tend to collapse when trained on a few samples [4] which can therefore generate many false positive alarms. In order to regularize the generalization capacity of the AE, DFR [47] shows that the integration of local and global information is a key point to improve existing AD methods. They train an AE to reconstruct multi-scale feature maps, which are themselves generated by concatenating different layers of a pre-trained CNN. Our proposed method follows this line of architecture design.

Distribution-based methods: An important trend is to use large networks on external training datasets such as ImageNet [15] to model the distribution of normal features. For example, P-SVDD [12], PaDiM [14], SPADE [13], PatchCore [40] or U-Students [7] assume that the normal data fits into a predefined kernel space. They then have to define the distances between the normal data and the abnormal data, which are assumed to be located outside this space. To do this, P-SVDD looks for the smallest hyper-sphere that surrounds the normal data and uses the Euclidean distances to the center of the hypersphere to detect anomalies [12]. While some models use clustering techniques [13], [40], [6] to detect samples outside the normal distribution of the data, others model this distribution by Gaussian models [14], [22], [7], [34].

These models [14], [12], [13], [40], [7], [40] perform better than reconstruction-based methods but have two main limitations: first, they work with patches of images, which leads to high complexity at the time of inference, second, to measure distances, they usually use flattened descriptors that destroy spatial positional relations of 2D images. In addition, these techniques may require the use of different patch sizes, as using patches that are too large may lead the model to ignore small abnormal areas and vice versa for patches that are too small [7].

In line with DFR [47], a number of AD methods, *e.g.*, Intra [38], SAAE [55] or PatchCore [40] or DifferNet [41], also explore the concept of multi-scale information gathering a step further, using Transformer or CNNs. Differnet has reused this idea of multi-scale feature maps, and implemented a normalization flow to detect anomalies [41].

Self-supervised learning based methods: It is now widely accepted that data augmentation strategies help to regularize CNN. To this end, various in-painting reconstruction methods have been developed in the context of AD [57], [38], [56]. For example Zavrtanik et al. showed that abnormal regions are less likely to be well reconstructed if they were not visible to the convolutional AE, and thus they treated this problem as a self-supervised reconstruction-by-inpaintings problem [56]. However at the inference time these methods suffer from high complexity since an anomaly map is generated via a set of in-painted versions of an input image. Many SSL-based methods have shown that the data augmentation strategy plays a critical role in defining an effective predictive task. For example, the application of basic geometric variations, such as rotations [25] or random affine transformations [5] performs poorly on texture images or symmetric objects [20], [5], [25] [31], that could be found found in the MVTec dataset [6].

Based on this claim, Cut&Paste [31] developed an SSL model in which the proxy task is adapted to detect irregular patterns. They created a data augmentation strategy in which a patch in an image is copied to another location after being randomly modified. This data-driven strategy outperforms the state of the art in terms of image-level classification. Nevertheless, at the time of inference, this method must use image patches to accurately locate anomalies.

Our HaloAE also leverages this data augmentation strategy to regularize the proposed HaloNet-based AE.

2.2 Visual Transformer

Transformer [53] has revolutionized natural language processing (NLP) tasks, such as translation [16], by allowing to model distant dependencies between elements of an input sequence and to parallelize sequence processing. In the last two years, many efforts have been made to adapt Transformer to computer vision tasks, such as image classification [18], [33], [39], object detection [9], [59], or image generation [11], [27], [36]. The self-attention mechanism, at the core of Transformer, is a memory- and computationally-intensive procedure. Therefore, to allow this operation on matrices, the Transformer kernel has been redesigned according to two main approaches: either by the global or a local computation of the self-attention [29].

Global approaches: Visual Transformer (ViT) is the first adaptation of Transformer to images [18]. The architecture of ViT is very similar to the original [53] but instead of taking a sequence of 1D words, 2D patches of an image are vectorized to feed a Transformer-like AE. However, this simple implementation requires a large dataset for training. Different approaches such as DeiT [50], CrossViT [10] or Criss-cross [26] have improved the initial performance of ViT, either by reducing the amount of training data required [50], or by taking into account more contextual information [26], [10].

However, all these methods require to decompose a 2D input into a sequence of vectors which implies two major limitations [18], [50], [10]: on the one hand this destroys intra-patch positional dependencies, and on the other hand, it does not allow the computation of correlations within patches.

Local approaches: Given these limitations, local versions of Transformer have been proposed. SASA proposed the first method based on a pure local self-attention model for images, and developed the computation of self-attention centered around each pixel via 2D grid extraction [39]. Although promising, this method lags behind the state of the art [52]. Recently, Swin Transformer has implemented a shifted window approach. It limits the self-attention operation to non-overlapping windows, while allowing connection between windows by merging neighboring patches into the deeper layers [33].

Vaswani et al. published HaloNet, where they proposed a block-based local self-attention that achieved the best speed/accuracy tradeoff for image classification tasks [52]. This result is obtained by violating the translation equivariance rule to obtain a better hardware utilization. Assuming that neighboring pixels share most of their neighborhood, HaloNet extracts a local neighborhood for a block of pixels in a single run. This *block-based* strategy allows parallelizing the self-attention operation. This technique improves both speed and memory management without affecting the performance, making the model more practical and hinting at its adaptation to larger widths [52].

2.3 Transformer for anomaly detection

Some papers have used the advantages of self-attention for unsupervised AD [58], [35], [55], [38]. For example, Zhang et al. integrated a multi-headed attention network between an AE that was adversely trained to reconstruct defect-free images [58]. VT-ADL uses ViT [18] as an encoder and a CNN as a decoder to reconstruct anomaly-free images, while a Gaussian mixture density network is implemented to refine the localization of anomalies [35]. Yang et al. implemented SAAE an AE based entirely on the self-attention mechanism, using ViT for both feature extraction, like DFR [47], and ViT as the AE for reconstructing these multi-scale feature maps. Pirnay et al. proposed a similar architecture with InTra, which is a purely self-attention approach based on reconstruction by inpainting. They showed that the inpainting scheme can be used to hide anomalous regions to further restrict the model's ability to reconstruct them [38]. However, these techniques using ViT suffer from its inherent limitations. The AD is therefore conditioned by the size of the patches, by the fact that intra-patch positional information is destroyed via vectorization, and by the fact that intra-patch correlations are not taken into account, *i.e.*, local information. In this work, we propose to leverage a block-based local attention, *i.e.*, HaloNet, to define our AE to acheive a single model for both anomaly detection and segmentation [52].

3 Method

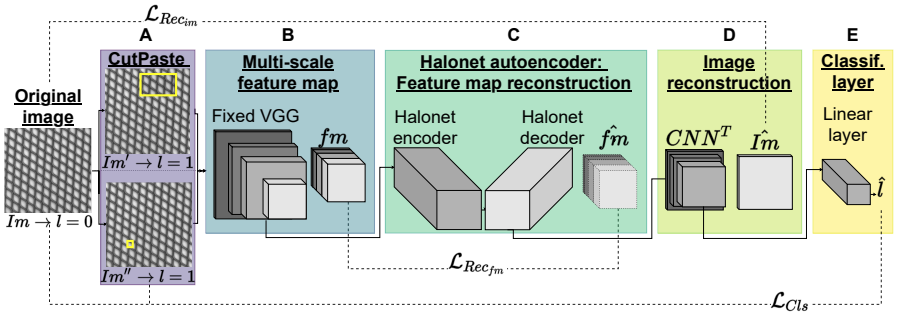

Fig. 2. Overview of HaloAE for AD. **A)** Cut&Paste data augmentation strategy for the SSL [31]. **B)** Multi-scaled feature map extraction via a pretrained VGG19 network [48] on ImageNet [15]. **C)** Halonet AE for feature map reconstruction. **D)** Reconstruction of images via transposed VGG blocks. **E)** Linear layer to determine the classification loss. Im and $\hat{Im}$, refer to the original image and the reconstructed image respectively, similarly fm and $\hat{fm}$ refer to the feature map and its reconstruction. l and $\hat{l}$ refer to the label and its prediction, 0 is associated to the original picture, and 1 to its augmented versions. $\mathcal{L}_{cls}$, $\mathcal{L}_{Rec_{fm}}$ and $\mathcal{L}_{Rec_{im}}$ refer to the classification loss and reconstruction quality of feature maps and images respectively.

3.1 Architecture

Self-supervised learning framework: To mimic industrial anomalies on the MVTec dataset [6] we re-used the strategy set up by Cut&Paste. This involves cutting out a rectangular patch, or any other shape according to the *scar* strategy of Cut&Paste. These elements of variable size and aspect ratio are cut from an input image and re-pasted at a random location, after undergoing random transformations such as rotations or color variations (Fig.2 - block A). This framework allows to define a proxy 2-ways classification task between normal and artificially damaged images. Let IM be the set of training images of size N such that $IM = \{im_0, ..., im_N\}$, where each im_i is in $\mathbb{R}^{w \times h \times c}$, with w, h and c the input width, height and number of channels. We define a classification loss function such as:

$$\mathcal{L}_{cls} = \frac{1}{N} \sum_{i=0}^{N} \mathbb{CE}(g(\hat{im}_i), l = 0) + \mathbb{CE}(g(CP(\hat{im}_i)), l = 1), \tag{1}$$

where the function $\mathbb{CE}(.)$ refers to the binary cross entropy function, $CP(.)$ to the Cut&Paste data augmentation strategy, and $g(.)$ to the binary classifiers, shown in Fig.2 - block E. This terminal linear layer takes as input a reconstructed image $\hat{im}_i$ associated with its label l , which is equal to 1 if the image has been augmented by $CP(.)$ and 0 otherwise.

Image features extraction: Following the DFR [47] method, we used a VGG19 [48] network trained on ImageNet [15] to extract the multi-scale features of an image. Given a deep CNN composed of L layers, it generates L feature maps, denoted $\{\Phi_1(im), ..., \Phi_L(im)\}$, with lower layer feature maps encoding low-level patterns such as texture with small receptive fields, while deep layer feature maps capturing higher-level information such as objectness with larger receptive fields. To bring all these pieces of multi-scale information together, we aggregate these different feature maps from $\{\Phi_l(im)\}_{l=1}^{L}$ to achieve a multi-scale feature map. Since VGG19 is composed of 16 convolution layers, we choose to concatenate the 1^{st}, 3^{rd}, 5^{th}, and 8^{th} layers of this network, which have a receptive field of 3, 10, 24, and 48 pixels, respectively. As reported by PathCore [40], we exclude features from very deep layers to avoid using overly generic features that are heavily biased towards ImageNet classification. The resulting multi-scale feature map is denoted $fm \in \mathbb{R}^{w_1 \times h_1 \times c_1}$, here w_1 and h_1 equal to 64 and c_1 to 704, (see Fig.2 - block B).

Reconstruction strategy grounded on Halonet: The self-attention operation captures distant relationships between pixels and generates spatially varying filters unlike convolutional layers [39], [52]. We make use of the block-based local self-attention introduced by HaloNet to create a reconstruction of fm denoted $\hat{fm}$. fm is divided into a grid of non overlapping blocks of size $\frac{h_1}{b}, \frac{w_1}{b}$. Every

block behaves like a group of query pixels. The haloing operation combines bands of hl pixels around each block to obtain the shared neighborhood from which the keys and values are calculated. In this way, the local self-attention per block multiplies each pixel in a shared neighborhood, after they have been transformed by the same linear transformation, by a probability considering both content-content and content-geometry interactions, resulting in spatially varying weights ([52] eq.2 and eq.3). In our study design we set the block size b to 12 and hl to 2, instead of using the original values which are 8 and 3 respectively, in order to capture more contextual information by taking advantage of the reduced size of the input since $h_1 = h/4$.

The architecture proposed by Vaswani et al. [52] is modified while keeping its ResNets-like structure [24]. Specifically, we have modified: (a) the head layer, substituting the 7x7 convolution with a stride of 2 by a 5x5 convolutional layer with a stride of 1, so as not to reduce the spatial dimension of the input map again; (b) the number of blocks per stage is set at 1 instead of the 3 or 4 in the original architecture, in order to create a lighter memory model; (c) in each block the second 1×1 convolution is replaced by a convolution layer with a filter of size 3×3 for the first two stages and 5×5 for the last two. This last modification allows both extracting local information with the 2D convolution layer. Increasing the filter size in the last two steps avoids the blocking effect created by HaloNet that could decrease the quality of the reconstruction (Table 1). It is important to note that we don't reduce the width and height of the input feature map since this reduction has already been performed by the pre-trained VGG19 network [48], therefore the HaloNet encoder learns a compressed version of the feature map fm by reducing its channels count. The encoded feature map fm_{enc} is in $\mathbb{R}^{60\times60\times58}$. From these encoded features fm is reconstructed by decoder combining both convolutional and block-local self attention layers. From these encoded features, fm is reconstructed by decoder combining both convolutional layers and local block self-attention layers. The decoder follows a similar architecture as the encoder, but all convolutional layers have been replaced by transposed convolutional layers, so we proposed the first transposed Halonet version.

The quality of the reconstructed feature maps is evaluated by a per-pixel loss L_2 and by a perceptual loss called the structure similarity index $SSIM$ [8]. Unlike L_2 which assumes independence between neighboring pixels, the $SSIM$ index evaluates the structural differences between the regions of the original and reconstructed maps by taking into account the co-variance between the regions [56], [35]. Therefore, the loss associated with feature map reconstruction is given by:

$$\mathcal{L}_{Rec_{fm}} = \sum_{i=1}^{h_1}\sum_{j=1}^{w_1} ||fm_{i,j} - \hat{fm}_{i,j}||_2 + (1 - SSIM(fm, \hat{fm}))_{(i,j)}, \qquad (2)$$

where the $SSIM$ is calculted between patches centered at (i, j).

To obtain a refined anomaly map at the image scale, we implemented a small transposed convolutional neural network, which is trained to reconstruct

Table 1. Summary of the HaloNet AE architecture: Each brace encloses a block, the number of blocks per stage is indicated in front of it. The batch normalization operation is denoted by BN, the convolution layers and the transposed convolution layers are denoted by conv and convT respectively. Finally, the number of channels at the end of each stage is indicated in the right-hand column for the encoder and decoder.

Halonet encoder		**Halonet decoder**	
5×5 conv, BN, relu	$d_h = 704$	$3 \times \begin{cases} 1 \times 1 \text{ conv}^{\text{T}}, \text{BN} \\ \text{Attention}(b,h), \text{relu} \\ 3 \times 3 \text{ conv}^{\text{T}}, \text{BN} \end{cases}$	$d_{dec_{s1}} = 29$ $d_{dec_{s2}} = 55$ $d_{dec_{s3}} = 118$
$2 \times \begin{cases} 1 \times 1 \text{ conv, BN} \\ \text{Attention}(b,h), \text{relu} \\ 3 \times 3 \text{ conv, BN} \end{cases}$	$d_{enc_{s1}} = 234$ $d_{enc_{s2}} = 117$	$1 \times \begin{cases} 1 \times 1 \text{ conv}^{\text{T}}, \text{BN} \\ \text{Attention}(b,h), \text{relu} \\ 5 \times 5 \text{ conv}^{\text{T}}, \text{BN} \end{cases}$	$d_{dec_{s4}} = 237$
$2 \times \begin{cases} 1 \times 1 \text{ conv, BN} \\ \text{Attention}(b,h), \text{relu} \\ 5 \times 5 \text{ conv, BN} \end{cases}$	$d_{enc_{s3}} = 58$ $d_{enc_{s4}} = 29$	$1 \times \begin{cases} 1 \times 1 \text{ conv}^{\text{T}}, \text{BN} \\ \text{Attention}(b,h), \text{relu} \\ 1 \times 1 \text{ conv}^{\text{T}}, \text{BN} \end{cases}$	$d_{dec_{s5}} = 704$

the input image im from $\hat{fm}$. It consists of five 2D convolution layers with filters of size 3×3, followed by a $ReLU$ activation function. Finally, a layer using the 2D nearest neighbor method is used to oversample the reconstructed image $\hat{im}$ to the scale of im (Fig.2 - block D). We use the same equation as eq.2 for the reconstruction of $\hat{im}$ as follows:

$$\mathcal{L}_{Rec_{im}} = \sum_{i=1}^{h} \sum_{j=1}^{w} ||im_{i,j} - \hat{im}_{i,j}||_2 + (1 - SSIM(im, \hat{im}))_{(i,j)}. \tag{3}$$

3.2 Loss function

By combining the losses described by eqs. 1, 2, and 3, we are defining a multi-objective problem. Usually the total loss $\mathcal{L}_T$ is written as a linear combination of the different losses $\mathcal{L}_i$ such that:

$$\mathcal{L}_T = \sum_i \alpha_i \mathcal{L}_i + \mathbb{R}(\boldsymbol{\alpha}), \tag{4}$$

where $\boldsymbol{\alpha}$ denotes a set of weights and $\mathbb{R}(.)$ some regularization of these weights. In general, the individual terms are weighted equally, assuming that each task contribute equally to the total loss, or $\boldsymbol{\alpha}$ weights are adjusted indivudally using an extensive grid search [21]. It has been shown that the exact value chosen for these weights can strongly affect the performance of the models [21] [19]. This can be explained by the fact that some losses might be in conflict, like in our case the classification loss and the reconstruction losses, while other can benefit from each other like the L_2 term and the SSIM term in our reconstruction equations (see eq.2 and eq.3). Inspired by the fact that humans often learn from an easy concept to a more difficult one, as pointed out by Li et al [30], we implemented

an adaptive weighting of the total loss function during learning. Therefore, the weighting of different $\mathcal{L}_T$ terms changes with the number of epochs t such that:

$$\mathcal{L}_T(t) = \alpha_1(t)\mathcal{L}_{cls} + \alpha_2(t)\mathcal{L}_{Rec_{fm}} + \alpha_3(t)\mathcal{L}_{Rec_{im}}. \quad (5)$$

We assume that the classification task is easier compared to the two reconstruction tasks, since it is a global decision at the image level while the quality of the reconstructions is evaluated at the pixel level. Moreover, since the quality $\hat{im}$ depends on the quality of $\hat{fm}$, we assume that $\mathcal{L}_{Rec_{fm}}$ must be optimized before $\mathcal{L}_{Rec_{im}}$. To this end, we modeled the evolution of α_1 by a decreasing logistic function, and α_2 and α_3 by two increasing logistic functions lagged by the number of epochs.

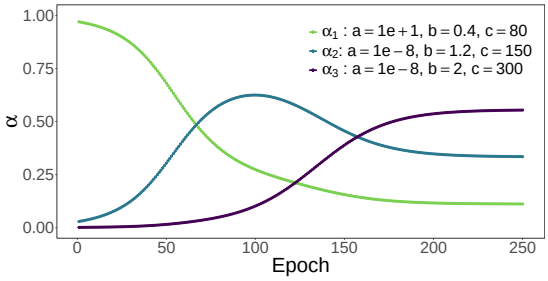

Fig. 3. Evolution of $\boldsymbol{\alpha}$ along with the number of epochs. Each curve is modeled according to the following equation, whose parameters are indicated in the legend: $\frac{(a-b)}{1+\exp(x-\frac{c}{2})^{0.05}} + b$. The $\boldsymbol{\alpha}$ values are then normalized so that they sum to 1.

4 Experiments

4.1 Experimental set-up

We evaluated our model on the recent challenging MVTec AD dataset [6]. The MVTec images were resized to 256×256 pixels. We applied data augmentation by randomly modulating the color. As explained above, each image is associated with two artificially damaged images using the Cut&Paste approach, either by copying and pasting a rectangle or a random shape [31].

All models have the same training hyperparameters: 250 training epochs, an Adam type optimizer with a learning rate of $1e^{-4}$ and a weight decay of $1e^{-5}$, the batch size is 12.

To assess our method, we calculated anomaly maps by comparing fm and its reconstruction $\hat{fm}$ via the L_2 distance such that:

$$A_{fm} = \sum_{i=1}^{h_1}\sum_{j=1}^{w_1} ||fm_{i,j} - \hat{fm}_{i,j}||_2. \quad (6)$$

To obtain an anomaly map from fm at the scale of im, we upsampled them by linear interpolation. Empirically, we observed that A_{fm} gives the best results in terms of both classification and localization tasks. The classification scores according to $\mathcal{L}_{cls}$ values and the segmentation scores obtained with A_{im} together with their associated anomaly maps resulting from image reconstructions, are given in supplementary Table S1.

The anomaly maps were post-processed to improve results (supplementary Fig. S1). They are first normalized by the average anomaly map of the training data and denoted A_{fm_N}. All the N anomaly maps from the training set A_{fm_j} are then concatenated before being averaged along the channel axis of dimension c_1. This operation reduces potential noise (supplementary Fig.S1). Finally, the anomaly maps are filtered using a Gaussian kernel of size 3×3, which smoothes the boundaries of the anomalous regions (Supplementary Fig.S1). Image-level AD is reported by the threshold-agnostic ROC-AUC metric. For the localization we reported the pixel-wise ROC-AUC.

4.2 Quantitative results

We compared our method with alternatives, ranging from AE L_2 [6], which can be considered the simplest model, to DifferNet [41], which uses the notion of likelihood through normalization flow to detect anomalies. We have also included Cut&Paste [31] and DFR [47] since we have reused parts of their method. Note that we recomputed the DFR results to have both the image-level AD ROC-AUC metric and the per-pixel segmentation ROC-AUC scores that are not available in the original paper. We also included P-SVDD [12] to refer to an embedding-similarity based method, as well as SAAE [55] and InTra [38] to refer to two other techniques using Transformer. The results are summarized in Table 2.

We can observe that HaloAE obtains satisfactory results for the detection of anomalies at the image level with an average ROC-AUC score of **91.4%**. This result is strongly influenced by the poor performance obtained specifically on the carpet object. As illustrated in Fig. S2, the network seems to be able to reconstruct the anomalies for this object, thus the distribution of A_{fm} means is similar between normal and abnormal objects. Notably, for some objects, image-level classification results are improved using the classification loss term $\mathcal{L}_{cls}$ (Table S1), although on average the post-processed anomaly map computed on the feature maps gives better results. For the pixel-wise segmentation results, HaloAE obtains an average ROC-AUC score of **91.2%**. This score is negatively impacted by the tile object, as shown in Fig.S3, HaloAE tends to detect the contours of anomalies. This can be explained by the local attention mechanism which calculates the interaction between neighboring pixels with respect to a central pixel.

It is important to note that P-SVDD [12], InTra [38] and Cut&Paste [31] are patch-based methods, either for learning and inference [12] or for inference only [31], [38]. Moreover, Cut&Paste uses two different models, one for classification and one for segmentation, while we proposed an all-in-one method. The preprocessing technique of InTra differs according to the objects, so this model

is not strictly unsupervised. Finally, SAAE does not report classification scores, which are not necessarily correlated with segmentation scores, as illustrated by our results on carpet.

Table 2. Anomaly detection and localization performance on the MVTec dataset. The first score in the pair refers to the image-level AD ROC-AUC score in percent, and the second to the pixel-wise ROC-AUC score in percent. The best score for each object is highlighted in bold.

	AE-l2	**P-SVDD**	**DFR**	**Cut&Paste**	**SAAE**	**InTra**	**DifferNet**	**HaloAE**
Carpet	(-, 59.0)	(92.9, 92.6)	(95.6, 98.5)	(**100.0**, 98.3)	(-, 97.9)	(98.8, **99.2**)	(84.0, -)	(69.7, 89.4)
Grid	(-, 90.0)	(94.4, 96.2)	(95.0, 97.4)	(96.2, 97.5)	(-, 97.9)	(**100.0**, **98.8**)	(97.1, -)	(95.1, 83.1)
Leather	(-, 75.0)	(90.9, 97.4)	(99.4, 99.3)	(95.4, 99.5)	(-, 99.6)	(**100.0**, **99.5**)	(99.4, -)	(97.8, 98.5)
Tile	(-, 51.0)	(97.8, 91.4)	(93.1, 90.9)	(**100.0**, 90.5)	(-, **97.3**)	(98.2, 94.4)	(92.9, -)	(95.7, 78.5)
Wood	(-, 73.0)	(96.5, 90.8)	(98.9, 95.4)	(99.1, 95.5)	(-, **97.6**)	(97.5, 88.7)	(99.8, -)	(**100.0**, 91.1)
Mean Text.	(-, 69.6)	(94.5, 93.7)	(96.4, 96.3)	(98.1, 96.3)	(-, **98.2**)	(**98.9**, 96.1)	(94.6, -)	(89.7, 88.1)
Bottle	(-, 86.0)	(98.6, 98.1)	(99.8, 95.8)	(99.9, 97.6)	(-, 97.9)	(**100.0**, **97.1**)	(99.0, -)	(**100.0**, 91.9)
Cable	(-, 86.0)	(90.3, **96.8**)	(78.9, 91.4)	(**100.0**, 90.0)	(-, **96.8**)	(70.3, 91.0)	(86.9, -)	(84.6, 87.6)
Capsule	(-, 88.0)	(76.7, 95.8)	(96.2, **98.5**)	(**98.6**, 97.4)	(-, 98.2)	(86.5, 97.7)	(88.8, -)	(88.4, 97.8)
HazelNut	(-, 95.0)	(92.0, 97.5)	(97.0, 92.0)	(93.3, 97.3)	(-, 98.5)	(95.7, **98.3**)	(91.1, -)	(**99.8**, 97.8)
MeatalNut	(-, 86.0)	(94.0, **98.0**)	(93.1, 93.3)	(86.6, 93.1)	(-, 97.6)	(**96.9**, 93.3)	(95.1, -)	(88.4, 85.2)
Pill	(-, 85.0)	(86.1, 95.1)	(91.9, 96.8)	(**99.8**, 95.7)	(-, 98.1)	(90.2, **98.3**)	(95.9, -)	(90.1, 91.5)
Screw	(-, 96.0)	(81.3, 95.7)	(94.3, **99.0**)	(90.7, 96.7)	(-, 98.9)	(95.7, **99.5**)	(99.3, -)	(89.6, **99.0**)
Toothbrush	(-, 93.0)	(100.0, 98.1)	(100.0, 98.5)	(97.5, 98.1)	(-, 98.1)	(**100.0**, **98.9**)	(96.1, -)	(97.2, 92.9)
Transistor	(-, 86.0)	(91.5, **97.0**)	(80.6, 79.1)	(**99.8**, 93.0)	(-, 96.0)	(95.8, 96.1)	(96.3, -)	(84.4, 87.5)
Zipper	(-, 77.0)	(97.9, 95.1)	(89.9, 96.9)	(**99.9**, **99.3**)	(-, 96.9)	(99.4, 99.2)	(98.6, -)	(99.7, 96.0)
Mean Obj.	(-, 87.8)	(90.8, 96.7)	(91.6, 94.4)	(**96.6**, 95.8)	(-, **97.7**)	(93.1, 96.9)	(94.6, -)	(92.2, 92.7)
Mean	(71.0, 82.9)	(92.1, 95.7)	(93.3, 95.1)	(**97.1**, 96.0)	(-, **97.9**)	(95.0, 96.7)	(94.9, -)	(91.4 91.2)

4.3 Qualitative results

We visualize some results of the anomaly localization in Fig. 1. The first and third rows show the input images while the second and last rows show the post-processed anomaly maps. These representations highlight that HaloAE is capable of locating tiny defects, as illustrated by the screw, capsule or zipper, and large defects, as illustrated by the hazelnut or the tile. In addition, HaloAE detects both structural defects, as shown by the wood and the tile, and color defects, as in the cable example, where the cable in the lower left corner is supposed to be red.

4.4 Ablation study

To study the effectiveness of the different modules of our workflow, we performed different ablation experiments exploring our loss function (eq.5) and the different blocks of our network (Fig.2). The results of the loss function modifications are summarized in Table 3. First, we highlighted the importance of adaptive weighting of the different $\mathcal{L}_T$ terms for the classification and segmentation tasks, with an average loss of 12.4 and 4.9 points respectively without it (2^{nd} row of Table 3). Considering this, we weighted $\mathcal{L}_T$ taking into account the homoscedastic uncertainty of each task, following the well-known strategy of Kendall et al. [28]. In the case the loss function is re-written as:

$$\mathcal{L}_T = \sum_{i=1}^{3} \frac{\mathcal{L}_i}{\sigma_i^2} + \sum_{i=1}^{3} \log(\sigma_i^2) \tag{7}$$

where each loss term is denoted by $\mathcal{L}_i$ and σ_i is the uncertainty parameter of each task. The results show that our weighting scheme is better for each of the two scores, emphasizing the importance of learning difficult tasks after easy ones (3^{rd} row of the Table 3).

We then showed the importance of image reconstruction module (Fig.2 - block D). The deletions of the transposed CNN associated with image reconstruction and the $\mathcal{L}_{Rec_{im}}$ term have a significant impact on the classification and segmentation scores with a decrease of 12.1 and 20.0 points respectively (4^{th} row of Table 3). Note the detrimental effect of this ablation on the segmentation score while we used the anomaly maps from the feature maps reconstruction to determine this score. The removal of the loss term associated with feature maps reconstruction penalizes the results with a decrease of 18.9 and 9.0 points for each of the two scores (5^{th} row of the Table 3). This effect is expected since the weights of the upstream VGG network remain fixed (Fig.2 - block B).

To evaluate the effect of the SSL module (Fig.2 - block A), we removed the loss term associated with the classification, as well as the data augmentation strategy. For this experiment, the adaptive weighting scheme had to be removed because it is mainly driven by $\mathcal{L}_{cls}$. As expected, these deletions had a strong impact on the classification results with a decrease of 14.6 points while the segmentation score remains stable with a loss of 3.4 points (6^{th} row of the Table 3). Nevertheless, if the model is trained only on classification loss, there is no improvement in classification scores[5] (7^{th} row of the Table 3). However, for this last experiment, the good classification results obtained on the texture objects suggest that the Cut&Paste strategy might be unstable in our architecture.

Table 3. Ablation study on loss function. The first row shows the final scores of our model, while the other rows highlight the effects of different $\mathcal{L}_T$ modifications. In each pair, the first element refers to the image-level AD ROC-AUC score and the second to the pixel-wise ROC-AUC score. The best score per column is highlighted in bold.

	Mean Text.	**Mean Obj.**	**Mean**
$\mathcal{L}_T(t) = \alpha_1(t)\mathcal{L}_{cls} + \alpha_2(t)\mathcal{L}_{Rec_{fm}} + \alpha_3(t)\mathcal{L}_{Rec_{im}}$	(89.7, **88.1**)	(**92.2**, 92.7)	(**91.4**, **91.2**)
$\mathcal{L}_T = \mathcal{L}_{cls} + \mathcal{L}_{Rec_{fm}} + \mathcal{L}_{Rec_{im}}$	(65.3, 74.2)	(86.7, **93.0**)	(79.0, 86.3)
$\mathcal{L}_T(t)$ equal to eq.7 uncertainty weighting [28]	(97.2, 87.9)	(82.53, 84.61)	(88.2, 85.9)
$\mathcal{L}_T(t) = \alpha_1(t)\mathcal{L}_{cls} + \alpha_2(t)\mathcal{L}_{Rec_{fm}}$	(83.1, 77.3)	(65.0, 80.1)	(71.43, 79.1)
$\mathcal{L}_T(t) = \alpha_1(t)\mathcal{L}_{cls} + \alpha_2(t)\mathcal{L}_{Rec_{im}}$	(74.3, 74.6)	(71.6, 86.0)	(72.5, 82.2)
$\mathcal{L}_T(t) = \mathcal{L}_{Rec_{fm}} + \mathcal{L}_{Rec_{im}}$	(63.9, 75.0)	(83.2, 94.2)	(76.8, 87.8)
$\mathcal{L}_T(t) = \mathcal{L}_{cls}$	(**97.4**, 59.0)	(75.6, 92.7)	(82.9, 70.6)
$\mathcal{L}_T(t) = \mathcal{L}_{Rec_{fm}}$	(62.1, 82.8)	(88.3, 94.1)	(79.6, 90.3)

[5] For this experiment the classifications scores were calculated in function of $\mathcal{L}_{cls}$ values.

To evaluate our network architecture, we first compared the performance of HaloNet as an AE, retaining only block C in Fig.2, to convolutional AEs trained via $\mathcal{L}_2$ or $\mathcal{SSIM}$ loss. In this experiment, our model is optimized to reconstruct images via the combination of $\mathcal{L}_2$ and $\mathcal{SSIM}$ losses. HaloNet as an AE does not perform as well as convolutional AEs, suggesting that our model is able to reconstruct abnormal regions through greater generalization ability (4^{th} row of Table 4). This justifies the need for the feature extractor module (Fig.2 - block B). Next, we replaced the HaloNet AE module with the convolutional AE from DFR [47]. In our architecture, the use of local block self-attention improves the results with an increase of 6.4 and 0.8 points for classification and segmentation respectively (5^{th} row of Table 4).

Table 4. Ablation study on the architecture. The first row shows the final scores of our model. In each pair, the first element refers to the image-level AD ROC-AUC score (in percent) and the second to the pixel-wise ROC-AUC score (in percent). The best score per column is highlighted in bold.

	Mean Text.	**Mean Obj.**	**Mean**
HaloAE (final)	(**89.7**, 88.1)	(**92.2**, **92.7**)	(**91.4**, **91.2**)
AE-l2	(70.0, 69.2)	(88.0, 88.9)	(82.0, 82.5)
AE-SSIM	(78.0, 78.2)	(91.0, 91.2)	(87, 86.9)
HaloAE - Block C only	(75.6, 67.4)	(78.2, 78.8)	(77.3, 75.0)
HaloaAE - Block C as CNN	(89.5, **94.1**)	(82.7, 90.4)	(85.0, 90.4)

5 Discussion and Conclusion

To the best of our knowledge, HaloAE is the first model to incorporate a local version of Transformer, along with HaloNet [52], to handle an AD problem. Computing intra-patch correlations via the local block self-attention operation improves both detection and localization. The module optimizing the oversampling of feature maps allows us to obtain an all-in-one model, which does not require an expansive patch-based process for anomaly segmentation. We also show that the integration of an SSL approach leads to a better regularization of the AE, ultimately improving the detection score at the image level. Finally, the improved scores brought by our new adaptive loss function weighting schemes suggest that learning multiple tasks simultaneously would be facilitated by giving increasing importance to the most difficult tasks.

Overall, HaloAE performed competitively on the MVTec dataset [6], with an average score of 91.4% for image-level detection and 91.2% for pixel-wise segmentation. The performances of our hybrid model between CNN and local Transformer suggests the importance of integrating global and local information at each step of the process. This study therefore implies that Transformer-based vision models could be improved by simultaneously applying the self-attention operation between and within patches [37] [33] [23].

Acknowledgments

This work for Liming Chen was in part supported by the 4D Vision project funded by the Partner University Fund (PUF), a FACE program, as well as the French Research Agency, l'Agence Nationale de Recherche (ANR), through the projects Learn Real (ANR-18-CHR3-0002-01), Chiron (ANR-20-IADJ-0001-01), Aristotle (ANR-21-FAI1-0009-01), and the joint support of the French national program of investment of the futur and the the regions through the PSPC FAIR Waste project. This work was granted access to the HPC resources of IDRIS under the allocation 2022-[AD011012172R1] made by GENCI.

References

1. Adam, A., Rivlin, E., Shimshoni, I., Reinitz, D.: Robust real-time unusual event detection using multiple fixed-location monitors. IEEE transactions on pattern analysis and machine intelligence **30**(3), 555–560 (2008)
2. Akcay, S., Atapour-Abarghouei, A., Breckon, T.P.: Ganomaly: Semi-supervised anomaly detection via adversarial training. In: Asian conference on computer vision. pp. 622–637. Springer (2018)
3. Akçay, S., Atapour-Abarghouei, A., Breckon, T.P.: Skip-ganomaly: Skip connected and adversarially trained encoder-decoder anomaly detection. In: 2019 International Joint Conference on Neural Networks (IJCNN). pp. 1–8. IEEE (2019)
4. Baur, C., Wiestler, B., Albarqouni, S., Navab, N.: Deep autoencoding models for unsupervised anomaly segmentation in brain mr images. In: International MICCAI Brainlesion Workshop. pp. 161–169. Springer (2018)
5. Bergman, L., Hoshen, Y.: Classification-based anomaly detection for general data. arXiv preprint arXiv:2005.02359 (2020)
6. Bergmann, P., Fauser, M., Sattlegger, D., Steger, C.: 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 (2019)
7. Bergmann, P., Fauser, M., Sattlegger, D., Steger, C.: Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4183–4192 (2020)
8. Bergmann, P., Löwe, S., Fauser, M., Sattlegger, D., Steger, C.: Improving unsupervised defect segmentation by applying structural similarity to autoencoders. arXiv preprint arXiv:1807.02011 (2018)
9. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision. pp. 213–229. Springer (2020)
10. Chen, C.F., Fan, Q., Panda, R.: Crossvit: Cross-attention multi-scale vision transformer for image classification. arXiv preprint arXiv:2103.14899 (2021)
11. Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., Sutskever, I.: Generative pretraining from pixels. In: International Conference on Machine Learning. pp. 1691–1703. PMLR (2020)
12. Cohen, N., Hoshen, Y.: Sub-image anomaly detection with deep pyramid correspondences. arXiv preprint arXiv:2005.02357 (2020)

13. Cohen, N., Hoshen, Y.: Sub-image anomaly detection with deep pyramid correspondences. arXiv preprint arXiv:2005.02357 (2020)
14. Defard, T., Setkov, A., Loesch, A., Audigier, R.: Padim: a patch distribution modeling framework for anomaly detection and localization. In: International Conference on Pattern Recognition. pp. 475–489. Springer (2021)
15. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
16. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
17. Dinh, L., Sohl-Dickstein, J., Bengio, S.: Density estimation using real nvp. arXiv preprint arXiv:1605.08803 (2016)
18. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
19. Dosovitskiy, A., Djolonga, J.: You only train once: Loss-conditional training of deep networks. In: International conference on learning representations (2019)
20. Golan, I., El-Yaniv, R.: Deep anomaly detection using geometric transformations. arXiv preprint arXiv:1805.10917 (2018)
21. Groenendijk, R., Karaoglu, S., Gevers, T., Mensink, T.: Multi-loss weighting with coefficient of variations. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1469–1478 (2021)
22. Gudovskiy, D., Ishizaka, S., Kozuka, K.: 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 (2022)
23. Han, K., Xiao, A., Wu, E., Guo, J., Xu, C., Wang, Y.: Transformer in transformer. Advances in Neural Information Processing Systems **34** (2021)
24. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
25. Hendrycks, D., Mazeika, M., Kadavath, S., Song, D.: Using self-supervised learning can improve model robustness and uncertainty. arXiv preprint arXiv:1906.12340 (2019)
26. Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., Liu, W.: Ccnet: Criss-cross attention for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 603–612 (2019)
27. Jiang, Y., Chang, S., Wang, Z.: Transgan: Two transformers can make one strong gan. arXiv preprint arXiv:2102.07074 **1**(3) (2021)
28. Kendall, A., Gal, Y., Cipolla, R.: Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7482–7491 (2018)
29. Khan, S., Naseer, M., Hayat, M., Zamir, S.W., Khan, F.S., Shah, M.: Transformers in vision: A survey. arXiv preprint arXiv:2101.01169 (2021)
30. Li, C., Yan, J., Wei, F., Dong, W., Liu, Q., Zha, H.: Self-paced multi-task learning. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
31. Li, C.L., Sohn, K., Yoon, J., Pfister, T.: Cutpaste: Self-supervised learning for anomaly detection and localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9664–9674 (2021)

32. Liu, Y., Li, C.L., Póczos, B.: Classifier two sample test for video anomaly detections. In: BMVC. p. 71 (2018)
33. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. arXiv preprint arXiv:2103.14030 (2021)
34. Liznerski, P., Ruff, L., Vandermeulen, R.A., Franks, B.J., Kloft, M., Müller, K.R.: Explainable deep one-class classification. arXiv preprint arXiv:2007.01760 (2020)
35. Mishra, P., Verk, R., Fornasier, D., Piciarelli, C., Foresti, G.L.: Vt-adl: A vision transformer network for image anomaly detection and localization. arXiv preprint arXiv:2104.10036 (2021)
36. Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, L., Shazeer, N., Ku, A., Tran, D.: Image transformer. In: International Conference on Machine Learning. pp. 4055–4064. PMLR (2018)
37. Patel, K., Bur, A.M., Li, F., Wang, G.: Aggregating global features into local vision transformer. arXiv preprint arXiv:2201.12903 (2022)
38. Pirnay, J., Chai, K.: Inpainting transformer for anomaly detection. arXiv preprint arXiv:2104.13897 (2021)
39. Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., Shlens, J.: Stand-alone self-attention in vision models. arXiv preprint arXiv:1906.05909 (2019)
40. Roth, K., Pemula, L., Zepeda, J., Schölkopf, B., Brox, T., Gehler, P.: Towards total recall in industrial anomaly detection. arXiv preprint arXiv:2106.08265 (2021)
41. Rudolph, M., Wandt, B., Rosenhahn, B.: 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 (2021)
42. Schlegl, T., Seeböck, P., Waldstein, S.M., Langs, G., Schmidt-Erfurth, U.: f-anogan: Fast unsupervised anomaly detection with generative adversarial networks. Medical image analysis **54**, 30–44 (2019)
43. Schlegl, T., Seeböck, P., Waldstein, S.M., Schmidt-Erfurth, U., Langs, G.: Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In: International conference on information processing in medical imaging. pp. 146–157. Springer (2017)
44. Schlegl, T., Seeböck, P., Waldstein, S.M., Schmidt-Erfurth, U., Langs, G.: Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In: International conference on information processing in medical imaging. pp. 146–157. Springer (2017)
45. Seeböck, P., Waldstein, S., Klimscha, S., Gerendas, B.S., Donner, R., Schlegl, T., Schmidt-Erfurth, U., Langs, G.: Identifying and categorizing anomalies in retinal imaging data. arXiv preprint arXiv:1612.00686 (2016)
46. Sheynin, S., Benaim, S., Polyak, A., Wolf, L.: Local-global shifting vision transformers (2021)
47. Shi, Y., Yang, J., Qi, Z.: Unsupervised anomaly segmentation via deep feature reconstruction. Neurocomputing **424**, 9–22 (2021)
48. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
49. Sultani, W., Chen, C., Shah, M.: Real-world anomaly detection in surveillance videos. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6479–6488 (2018)
50. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: International Conference on Machine Learning. pp. 10347–10357. PMLR (2021)

51. Vasilev, A., Golkov, V., Meissner, M., Lipp, I., Sgarlata, E., Tomassini, V., Jones, D.K., Cremers, D.: q-space novelty detection with variational autoencoders. In: Computational Diffusion MRI, pp. 113–124. Springer (2020)
52. Vaswani, A., Ramachandran, P., Srinivas, A., Parmar, N., Hechtman, B., Shlens, J.: Scaling local self-attention for parameter efficient visual backbones. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12894–12904 (2021)
53. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
54. Venkataramanan, S., Peng, K.C., Singh, R.V., Mahalanobis, A.: Attention guided anomaly localization in images. In: European Conference on Computer Vision. pp. 485–503. Springer (2020)
55. Yang, Y.: Self-attention autoencoder for anomaly segmentation (2021)
56. Zavrtanik, V., Kristan, M., Sko caj, D.: Reconstruction by inpainting for visual anomaly detection. Pattern Recognition **112**, 107706 (2021)
57. Zavrtanik, V., Kristan, M., Skočaj, D.: Draem-a discriminatively trained reconstruction embedding for surface anomaly detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8330–8339 (2021)
58. Zhang, Y., Gong, Y., Zhu, H., Bai, X., Tang, W.: Multi-head enhanced self-attention network for novelty detection. Pattern Recognition **107**, 107486 (2020)
59. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)

Supplementary material

Table S1. Exploring the outputs of HaloAE and the post-processing procedure. The first pair score corresponds to the image-level AD ROC-AUC score in percent, and the second to the pixel-level ROC-AUC score in percent. The best score for each object is highlighted in bold. A represents the anomaly map, noted A_{im} or A_{fm} if calculated on the images or on the features map, respectively (see eq.6 in main text). $A_{\{im,fm\}_N}$ represents the normalized anomaly map. Finnaly, $\mathcal{N}_{filter}$ refers to the Gaussian filter applied to the normalized anomaly map.

	$\mathcal{L}_{cls}$	$A_{im_N} + \mathcal{N}_{filter}$	A_{fm}	A_{fm_N}	$A_{fm_N} + \mathcal{N}_{filter}$
Carpet	(56.9, -)	(**74.4**, 60.7)	(54.3, 88.1)	(60.7, **88.5**)	(69.7, 89.4)
Grid	(**100.0**, -)	(82.1, 53.4)	(94.5, 82.7)	(95.2, 83.0)	(95.1, **83.1**)
Leather	(71.0, -)	(60.2, 78.3)	(97.2, 98.0)	(**97.8**, 98.1)	(**97.8**, **98.5**)
Tile	(51.5, -)	(92.6, 66.1)	(93.3, 75.9)	(95.2, 76.1)	(**95.7**, **78.5**)
Wood	(93.2, -)	(99.0, 77.4)	(99.7, 90.7)	(99.9, 90.3)	(**100.0**, **91.1**)
Mean Text.	(74.5, -)	(81.66, 67.2)	(87.8, 87.1)	(**89.8**, 87.2)	(89.7, **88.1**)
Bottle	(98.4, -)	(99.9, 86.7)	(99.9, 90.0)	(**100.0**, 91.7)	(**100.0**, **91.9**)
Cable	(**100.0**, -)	(62.8, 76.3)	(79.2, 77/9)	(84.6, 86.1)	(84.6, **87.6**)
Capsule	(**96.8**, -)	(54.5, 63.6)	(83.2, 97.3)	(88.4, 97.4)	(88.4, **97.8**)
HazelNut	(99.4, -)	(86.3, 76.0)	(98.9, 97.9)	(99.6, 97.7)	(**99.8**, **97.8**)
MeatalNut	(**98.0**, -)	(65.2, 69.2)	(85.6, 86.3)	(88.4, 84.5)	(88.4, **85.2**)
Pill	(**100.0**, -)	(50.8, 77.2)	(86.4, **92.8**)	(90.6, 89.9)	(90.1, 91.5)
Screw	(**100.0**, -)	(54.6, 78.5)	(88.6, 98.8)	(89.6, 98.6)	(89.6, **99.0**)
Toothbrush	(58.1. -)	(89.7, 81.0)	(94.7, 93.0)	(**97.2**, 92.6)	(**97.2**, **92.9**)
Transistor	(**92.3**, -)	(81.5, 79.9)	(80.0, 84.8)	(84.4, 85.6)	(84.4, 87.5)
Zipper	(51.4, -)	(**99.7**, 86.8)	(99.7, 95.4)	(**99.7**, 95.3)	(**99.7**, **96.0**)
Mean Obj.	(89.4, -)	(74.5, 77.5)	(89.6, 91.6)	(**92.3**, 91.9)	(92.2, **92.7**)
Mean	(84.4, -)	(76.9, 74.1)	(89.0, 90.0)	(**91.4**, 90.4)	(**91.4**, **91.2**)

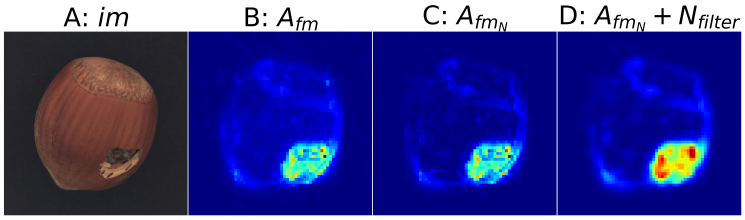

Fig. S1. Post processing workflow. **A)** Input image. **B)** Anomaly map (see eq.6 in main text). **C)** Normalized anomaly map (see eq.7 in main text). **D)** Normalized anomaly map smoothed with a Gaussian filter.

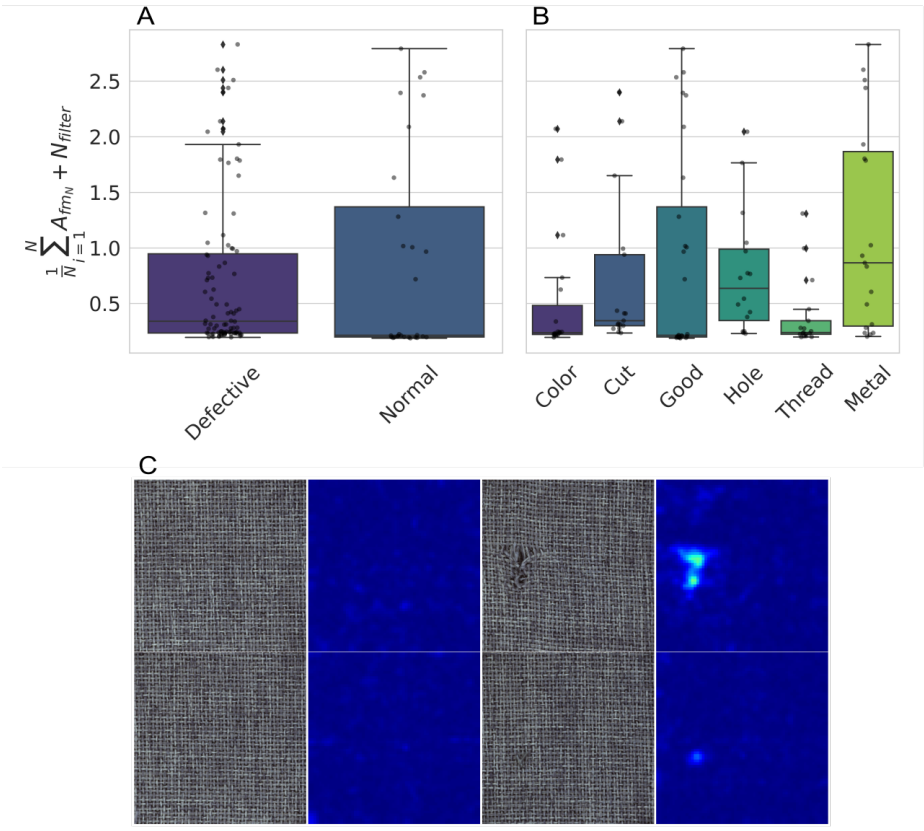

Fig. S2. Classification results on carpet. **A)** Distribution of means of post-processed anomaly maps computed on the feature map, for defect free and abnormal objects. **B)** Distribution of means of post-processed anomaly maps computed on the feature map by defect category. Defect-free objects and anomalous objects have similar distributions. **C)** Carpet anomaly map, on the left objects without defects, on the right abnormal objects.

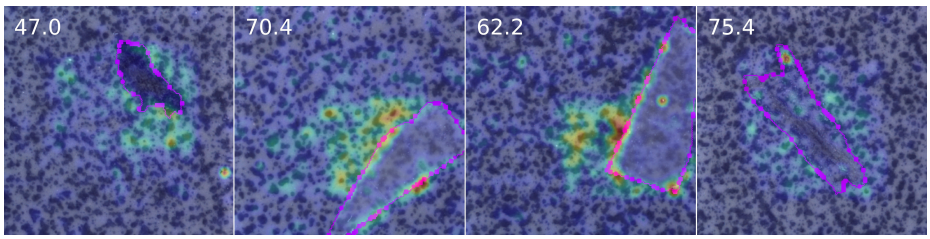

Fig. S3. Segmentation results on tiles. Each image shows the segmentation anomaly maps computed on the feature maps, the ground truth location is surrounded by a pink line. For each image, the anomaly score per pixel in percentage is shown in the upper left corner.