Digit. Signal Process. 168 (2026) 105613

Contents lists available at ScienceDirect

Digital Signal Processing

journal homepage: www.elsevier.com/locate/dsp

Scaling anomaly detection with segmentation models

Stefano Samele *, Francesco Attorre, Matteo Matteucci

Department of Electronics, Information and Bioengineering, Politecnico di Milano, Via Giuseppe Ponzio 34/5, Milan, 20133, Italy

ARTICLE INFO

Keywords:
Anomaly detection
Segmentation models
Unsupervised learning

ABSTRACT

Detecting anomalies in product images is a critical task in industrial quality control, where even subtle defects can have operational and financial impact. However, deploying anomaly detection algorithms in real-world industrial scenarios remains challenging, particularly when products are large, complex, or captured at high resolution. Many existing methods struggle to scale effectively while maintaining precision. This work aims to develop an algorithm that can effectively scale to larger, more complex objects. The method, *SADSeM* (Scaling Anomaly Detection with Segmentation Models), is based on classic convolutional neural networks for segmentation, such as Mask-RCNN. Thanks to these models' ability to learn and encode an object's structure, we can design a pipeline that uses both their segmentation maps and feature embeddings to carry out unsupervised anomaly detection. As the segmentation task is effectively solved by these models independently of image size, we scale to higher-resolution images with more effectiveness than competitors, while maintaining competitive results in simpler scenarios.

1. Introduction and related works

Anomaly detection fundamentally entails identifying instances within a dataset that deviate from expected or normal behavior. In industrial settings, anomaly detection is a pivotal tool for quality control assessments. Recent advances in computer vision, facilitated by deep learning algorithms, have illuminated the potential for fully automated quality control processes by analyzing camera-captured images. However, this endeavor encounters several challenges, including the possible complex nature of the object to explore. Many exciting algorithms have been proposed in recent years based on unsupervised learning: [23] and [14] offer extensive overviews of existing approaches. Traditionally, Anomaly Detection (AD) relied on reconstruction techniques, such as AutoEncoders (AE, [6,33]) or Generative Adversarial Networks (GAN, [1,30,34]) or Diffusion Models (DM [39]), which learn to compress and reconstruct images. However, neural networks often approximate anomalous features [9,36], making reconstruction error unreliable for detection. To address this issue, some studies aim to diminish reconstruction accuracy in anomalous regions through auxiliary tasks or regularization methods [2,41]. An alternative approach utilizes Normalizing Flows (NF, [25]), transforming simple distributions like Gaussians into more complex ones through invertible transformations. This enriches density estimation of conformant sample distributions, facilitating outlier detection [38]. DifferNet [27], for instance, employs NF to learn the density distribution of image features from a pre-trained Convolutional Neural Networks (CNNs). A subsequent iteration of this model, Fully Convolutional Cross-Scale Flow [28], preserves the image's 2D structure and compares features extracted at various network layers. Other AD methods introduce synthetic defects to augment the model's ability to discern between normal and abnormal examples in a semi-supervised setting [19,42]. For example, MEMSEG [40] trains a U-Net-based architecture to segment generated anomalous examples, starting from a pre-trained ResNet-18 [15]. Knowledge distillation (KD, [29]) involves transferring knowledge from a larger model to a smaller one, with the discrepancy between the two representations indicating anomaly presence. In a *reverse distillation* paradigm, as introduced in [11] and [17], the student network takes the teacher model's one-class embedding as input, trying to restore the teacher's multiscale representations, projecting high-dimensional features into a compact embedding space to highlight abnormal information.

The most effective techniques to date are embedding-based methods, leveraging CNNs like ResNet or EfficientNet [32], pre-trained on extensive datasets like ImageNet [13]. These methods extract activation maps from specific convolutional layers at different resolutions to create image embeddings. Among them, PaDiM [10] utilizes layers at various depths to capture both detailed information and global context, modeling embedding vectors as multivariate Gaussian distributions and estimating their mean and covariance matrix using normal samples. During testing, the Mahalanobis distance is computed between each test image patch and the estimated distribution. Coupled-hypersphere-based

* Corresponding author.
E-mail addresses: stefano.samele@polimi.it (S. Samele), francesco.attorre@studenti.polimi.it (F. Attorre), matteo.matteucci@polimi.it (M. Matteucci).

https://doi.org/10.1016/j.dsp.2025.105613

Available online 20 September 2025

Feature Adaptation for target-oriented anomaly localization (CFA, [18]) is a work that adapts features extracted by a pre-trained network using a learnable patch descriptor that embeds them into a set independent of the target dataset's size. PatchCore [26], to the best of our knowledge, stands as the most performant model among embedding-based methods, collecting spatial activations from a set of normal images and storing them in a memory bank. A greedy reduction procedure is applied to obtain a core set of features, with the distance to the nearest neighbor patch in the memory bank serving as an anomaly score. More recently, Position and Neighborhood Information (PNI, [4]), and EfficientAD [5] tried to address the limitations of such methods further. The first method involves predicting the normal distribution of features through conditional probability based on neighborhood characteristics, employing a multi-layer perceptron network for modeling. Additionally, positional data is incorporated by generating a histogram of characteristic features for each position. The second method combines the student-teacher approach (a typical procedure in Deep Learning, where a bigger model guides the training of a smaller one) to train a student network to predict features extracted from a teacher. The authors propose a training loss that prevents the student from imitating the teacher's feature extractor beyond the normal images. Even after these recent advancements, we are still far from widespread adoption of such algorithms in real-world industrial scenarios. This work is motivated by the ambition to provide a solution engineered to handle images of varying sizes and complexities, and to establish the link between the semantic segmentation task and the anomaly detection task. Our approach surpasses state-of-the-art (SOTA) methods on wide images in contexts constrained by memory limitations. Our method evolves from methods based on pre-trained convolutional neural network embeddings, with our original contributions encompassing:

- **A novel algorithm.** An innovative algorithm based on classical semantic segmentation models. We introduce a novel feature selection procedure to properly localize and select valuable features to describe the concept of normality in an object. We also develop a scoring algorithm based on two information types: the features themselves and the segmentation masks.
- **Pipeline coherence.** A method with each constituent element having a discernible influence on the overall pipeline. This characteristic renders our approach adaptable to diverse domains and scenarios with ease.
- **Scalability.** We demonstrate effective scalability to larger images and exhibit promise in handling data of great complexity.

2. Method

2.1. Semantic segmentation for anomaly detection

The task of recognizing the anomaly in an image representing an object is interconnected with understanding an object's essential parts. The process of identifying the object and its basic components inspired us to develop a pipeline based on segmentation models. The general idea is to have an organized and structured representation of an object and then, at inference time, try to compare what the model sees with what it has seen in the past. This approach should be beneficial when dealing with objects made of different complex parts potentially not enclosed in a small squared image representation. When the image resolution grows significantly, the model must group different semantically-linked portions of the objects. Our pipeline will face two essential problems: the localization of the object's components and the extraction of the correct features representing those parts. The method is based on three different parts. The first one adopts a fine-tuned segmentation model that takes the object image as input and outputs a segmentation mask of its relevant component with its corresponding bounding boxes; the second one involves a features localization and selection step able to extract from the same segmentation model the best internal representation of the object component; last, a scoring algorithm leverages both the features extracted at the previous step and the segmentation masks obtained in the first step.

2.2. MaskRCNN finetuning

Our anomaly detection pipeline initiates with the crucial step of employing a fine-tuned segmentation model, for which MaskRCNN [16] was selected. This decision is motivated by several key factors. First, the performance: until recent advancements, MaskRCNN stood as the state-of-the-art solution for semantic segmentation and object detection. Second, the MaskRCNN backbone, typically a pre-trained CNN for classification like ResNet or WideResNet, gives the chance to leverage the setting of pre-trained CNNs-based methods for Anomaly Detection. Last, the task of fine-tuning a segmentation model with few examples has been extensively addressed for MaskRCNN and other Fast and Faster R-CNN [31,35,37]. An easy and reliable method is thus required to fine-tune the chosen segmentation model without manually annotating many images to re-train it from scratch, since many AD benchmark datasets lack ground truth masks of the normal objects' components. Inspired by the approach outlined in [35], we fine-tune MaskRCNN on a limited set of samples using a pre-trained network. For each class in each used dataset, we tagged the whole object, from its external profile to its minor components adhering to a hierarchical tagging process. These levels, identified as sub-classes, are tailored to the diverse objects present in benchmark datasets, ranging from objects with singular tags to those with different nested tags. Starting with a dataset of images $I_k, k : 1 \dots C$, where C represents the number of classes, we annotate 10 images for each class $j_k, j : 1 \dots N_k$, with N_k representing the number sub-classes for class k. We deviate from the methodology detailed in [35] by unfreezing all network weights. This decision significantly influences subsequent stages by fostering the construction of more stable and consistent features. This fine-tuning process incorporates additional knowledge into the anomaly detection pipeline and enables the removal of extraneous image portions, such as background elements. Furthermore, domain experts can identify and exclude specific irrelevant parts, enhancing the pipeline's efficacy, mainly when dealing with high-resolution images. It also creates a relevant difference between normal and anomalous images. The model fails the segmentation from time to time, according to the anomaly it faces. We inserted in Fig. 1 some examples of the segmentation model outputs for some defective images of MVTec Anomaly Detection dataset (MVTec AD, [7]). The anomaly alters the segmentation abilities of the model. Thus, considering the distance between a set of correctly segmented components and those coming from an image at inference time, as we do in the anomaly scoring procedure, is essential. The presence of an anomaly shrinks and enlarges the segmentation mask or has no effect, providing critical information to create the final anomaly map.

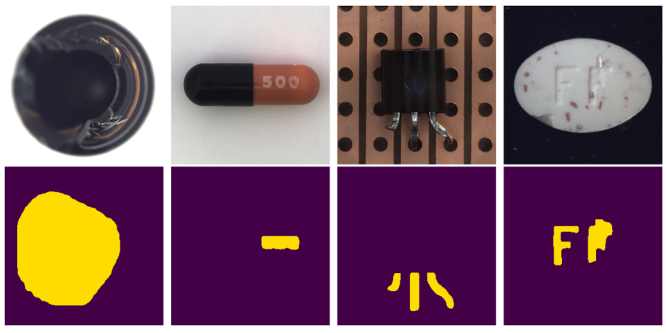

Fig. 1. Examples of interaction between the segmentation abilities of the model and the presence of anomalies.

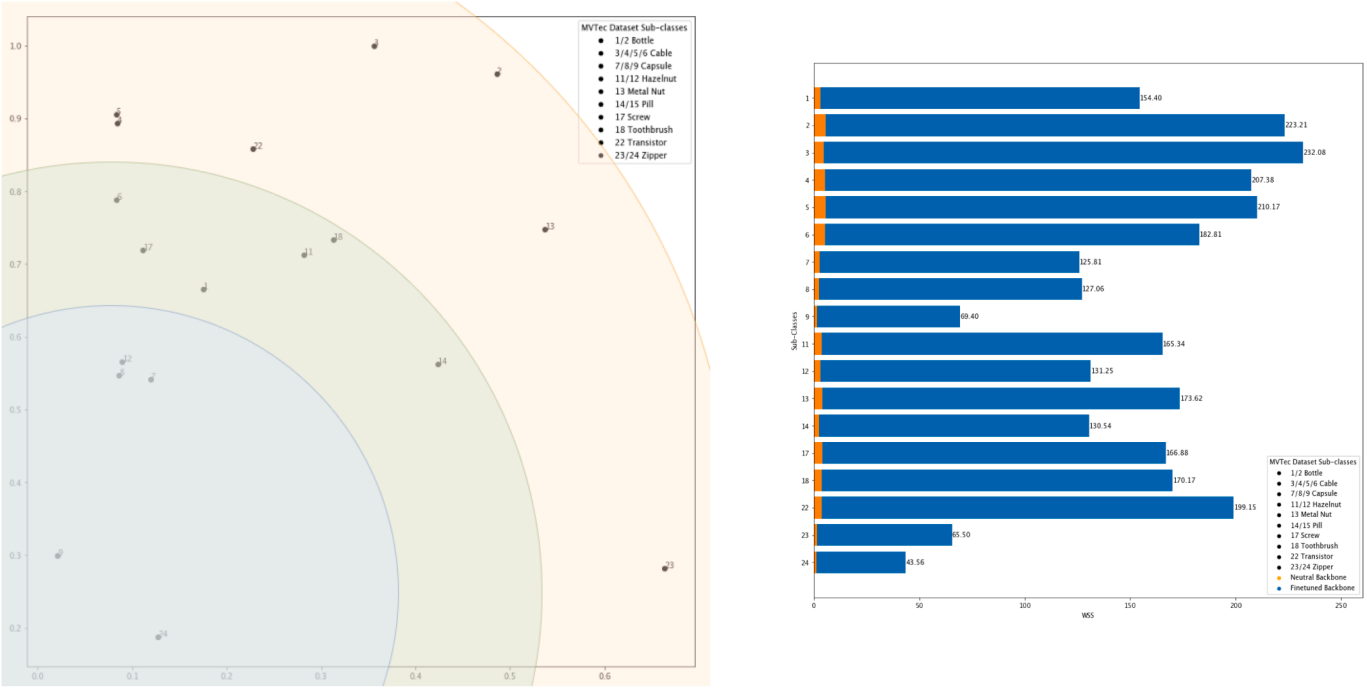

Fig. 2. On the left, different sub-classes of MVTec Anomaly Detection Dataset WSS plotted against their percentage area. The blue area identifies sub-classes that benefit from a representation through Layer 1; the green area represents those that benefit from a representation through Layer 1 and 2; the red area represents those that benefit from a representation through Layer 2; all this for 256x256 images. On the right, the same subclasses WSS values. The blue bars identify scores generated by extracting features from a fine-tuned backbone; the orange bars identify scores generated by extracting features from a non-fine-tuned (*neutral*) backbone.

2.3. Features localization and extraction

A model that can segment normal objects correctly can also be used to gather information about non-defective images to build a dictionary describing the normality concept. As highlighted in Section 1, CNN embedding-based methods have achieved promising results in the field of Anomaly Detection. In the same way, we collect spatial points from embeddings belonging to specific sub-classes from the internal representation of MaskRCNN. Thus, two critical aspects of our method are:

- Correctly locating the identified area in the embedding space;
- Picking one or more layers to build a features' set representing the expected behavior of a specific sub-class.

Inside MaskRCNN, there are two operations designed to execute the localization and extraction steps. The *level mapper* and the *RoI Align* operation. The first one assigns proposals coming from the Region Proposal Network (RPN, [24]) of specific dimensions to specific layers of the Feature Pyramid Network (FPN, [20]). The second one is introduced to correctly align the extracted features with the input. It produces better segmentation masks, aligning the prediction proposals in the features' space. During the training step of our method, for an image I_k of class k, the segmentation mask $M_{k,j}$ produced for a subclass j is considered. Using $M_{k,j}$, we extract from the backbones the most useful features coming from the activation maps of different layers in order to describe the concept of normality for subclass j. To achieve such a result, we perform two operations: the *mask projection* and the *layers assignment*. We project the segmentation mask back to the embeddings generated by the MaskRCNN backbone through the RoiAlign operation. This way, we can better localize the projection of a particular component of an object. Once we have identified the correct corresponding portion of activation maps $A_{k,j}(x : y)$, we apply a rescaled version of the obtained segmentation mask to the region, and we collect only those spatial points belonging to the mask. These have been thought to be a complete partition of the whole object; thus, each pixel of the original image only has a unique feature representation through a patch collected from specific layers.

After the localization procedure, we developed a strategy to pick the best features. Using the normal images, we have collected two fundamental statistics: the within-sum-of-squares (WSS) of Layer 2 and Layer 3 of WideResNet-50 (as they are generally considered the best layers to extract features from [4,26,28] normalized according to the number of points collected, and the average percentage dimension of the segmentation maps. Analyzing these data, we proposed the following strategy to assign each sub-class to different layers of the backbone.

We have collected the features separately for each sub-class of each class of MVTec AD dataset, and we have resampled the features according to the procedure described in [26]. We have plotted the WSS normalized per the number of points against the area. The normalized-WSS measures the spread of the points with respect to the cluster centroid; the more spread the features are, the more they reflect different aspects of the object component, thus the more complex the component is. We have identified that sub-classes with a low area and a low complexity (measured as normalized-WSS) benefit from a representation coming from lower layers. Moreover, we have identified an area-complexity ratio that we follow in order to assign each subclass to a specific layer. We hypothesize that if a cluster is made by a more spread or complex component, deeper features are more suitable to represent it. Fig. 2 (left) shows that, if the sub-class lies in a portion of the graph close to the origin, the model adopts Layer 1. Instead, when the complexity and the area of the segmentation maps of the sub-class increase, the model adopts both Layer 1 and Layer 2 or simply Layer 2. Also, features coming from a fine-tuned MaskRCNN are *over-specialized*, with respect to the non-fine-tuned version. Fig. 2 (right) shows that the features of the different sub-classes have a normalized WSS with an order of magnitude higher in the case of the fine-tuned backbone. This has a positive impact, as lower layers (1–2) gain the ability to better describe the features of an object and its components without compromising the resolution. This means that lower layers have features that better represent smaller/simpler regions. When the image grows in size while the dimension of the crops remains fixed, each component will either: a) have a more complex representation due to the zoom effect on specific details; b) occupy a larger area

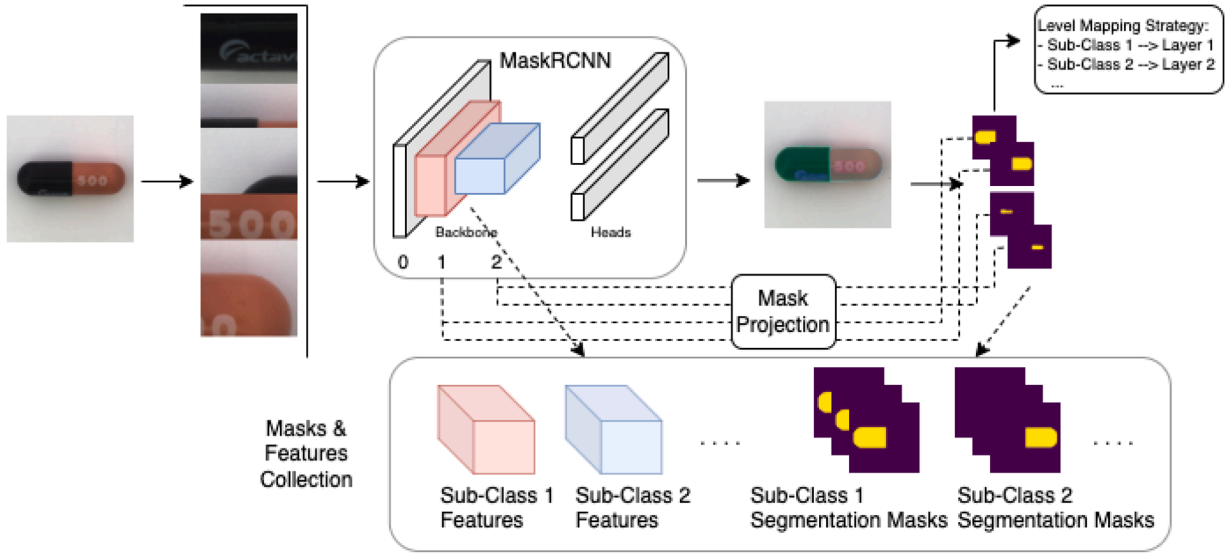

Fig. 3. A diagram of the features and masks' localization and collection procedure.

of the sub-crop. Thus, it will move from the origin to the outer regions of Fig. 2.

If this happens, more specialized layers will be used to describe the normality concept: features coming from Layer 2 or Layer 3 will be adopted. Once we have assigned to each sub-class the backbone layers the features should come from, we apply the mask projection and collect them, additionally applying the resampling procedure presented in [26].

2.4. Segmentation maps as additional information

As highlighted in sub-Section 2.2, the segmentation masks also carry relevant information to spot anomalies. Thus, we have collected and stored the segmentation masks M_j obtained directly from MaskRCNN for each sub-class. These masks do not have to be stored at full resolution, but we can conveniently resize them to a lower fixed dimension. To avoid false signals, for each collected segmentation map, we compute the distance to its nearest neighbor in the collection set and store it as a rescaling factor $RFM_{k,j}$. This simple solution has two advantages: first, we take into account small changes in the segmentation map of the object components; second, if we need to re-balance the importance between the features anomaly distance matrix and the segmentation anomaly distance matrix, we can simply downscale and store a resized version of the second.

At the end of both features and masks' extraction procedure, we are left with a collection of salient features $\boldsymbol{A}$ and segmentation masks $\boldsymbol{M}$. Fig. 3 represents a schematic diagram of this step.

2.5. Scoring algorithm

As the learning procedure leverages two types of information, so does the scoring algorithm. Once we have built a set of features able to describe the concept of normality for an object and its components, we produce an anomaly map of the input image, as well as an aggregated score.

When a new image $\hat{I}_k$ of class k comes at test time, it is pre-processed as describe above, to extract features $\hat{A}_{k,j(l)}(x,y)$ for each sub-class j, as well as its masks $\hat{M}_{k,j}$. x, y represent the activation map spatial positions. Each sub-class j depends on layer l through our layer mapping system. We then run a 1-Nearest-Neighbor search through the collection of features $\boldsymbol{A}$ to retrieve the closest element for $\hat{A}_{k,j}(x,y)$:

$$A^*_{k,j(l)}(x,y) = \text{argmin}_{A_{j(l)} \in \boldsymbol{A}} \|\hat{A}_{k,j(l)}(x,y) - A_{j(l)}(z,w)\| \quad (1)$$

where (z,w) are all the collected spatial points for subclass j through layer l, and $\|\cdot\|$ is the L_2 norm. The L_2 distance of $A^*_{k,j(l)}(x,y)$ and $\hat{A}_{k,j(l)}(x,y)$ for each x, y creates the anomaly distance matrix. For each subclass j, this distance matrix has a resolution equal to the selected layer l activation maps, but only points effectively belonging to the mask contribute to the calculation. Features' anomaly maps from different layers could potentially refer to different object components. Because the segmentation adopted is a partition of the whole object, the final anomaly map is assembled by merging the anomaly maps of different components, upsampling them to the input image resolution, and convolving them with a Gaussian smoothing filter. The image-level score is produced by simply considering the maximum of the unscaled anomaly maps produced by different components at different layers.

Also for the mask we run a 1-Nearest-Neighbor search through the collection of segmentation maps $\boldsymbol{M}$ to retrieve the closest element $\hat{M}_{k,j}$:

$$M^*_{k,j} = \arg \min_{M_j \in \boldsymbol{M}} |\hat{M}_{k,j} - M_j| \quad (2)$$

where this time $||$ is the L_1 norm. Once we have this element, the anomaly matrix is created by multiplying $|\hat{M}_{k,j}(x,y) - M^*_{k,j}(x,y)|$ by the distance from its nearest neighbor, i.e. the number of pixels the two masks differ: $|\hat{M}_{k,j} - M^*_j|$. To avoid false signals, for each collected segmentation map, we compute the distance to its nearest neighbor in the collection set and store it as a rescaling factor $RFM^*_{k,j}$. This value is subtracted from the mask-related anomaly map, and the result is clipped to zero.

Finally, the feature anomaly matrix and the segmentation anomaly matrix are min-max normalized inside each sub-class and aggregated considering the maximum function, both for image-level and pixel-level scores. The min-max normalization performed inside each subclass separately highlights the possible different dynamics of each component: the distance between normal and abnormal features can have different scales and ranges of separations according to the portion of the object they belong to. Fig. 4 represents a schematic view of this step.

3. Experiments

This section presents the experimental evaluation of our proposed anomaly detection methodology. We have designed and executed experiments to validate the efficacy and robustness of our approach across diverse industrial contexts and scenarios. The first set of experiments relates to the Mask-RCNN fine-tuning process. The second is an ablation

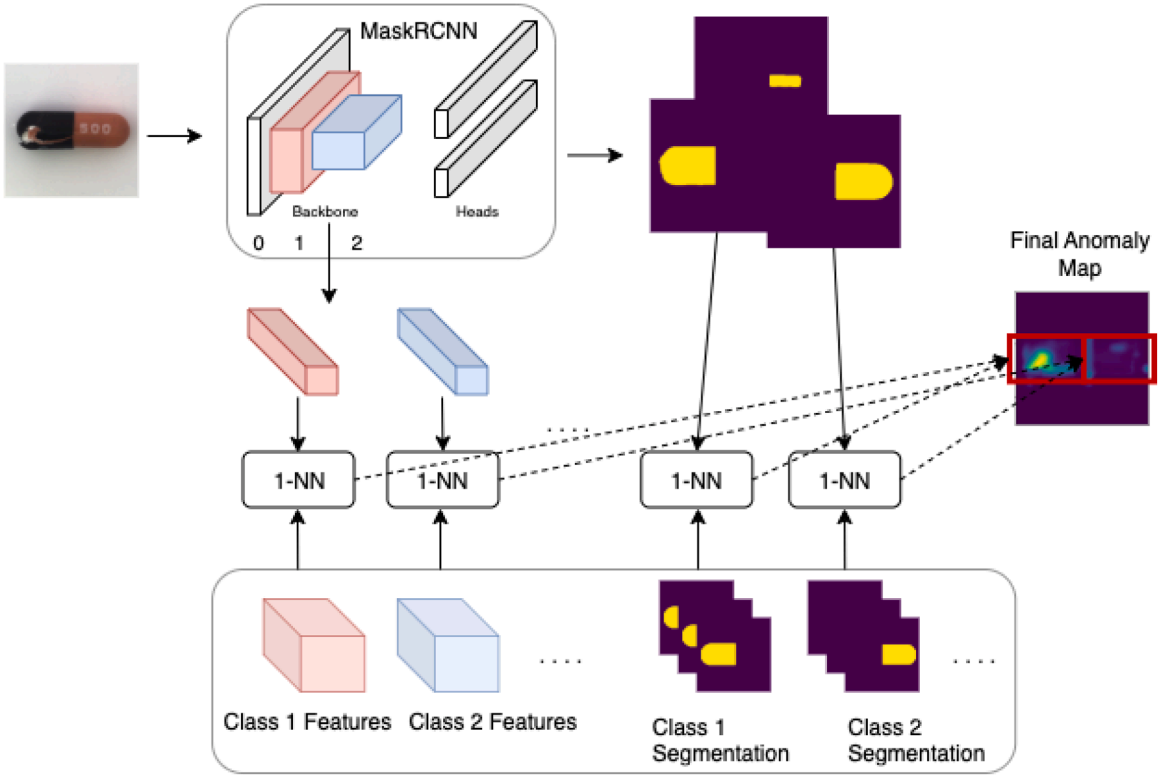

Fig. 4. A diagram of the anomaly scoring procedure.

study to evaluate the positive impact of the different pipeline building blocks.

In the third set of experiments, we have compared our performance against SOTA methods in anomaly detection over images. The last tranche of experiments has processed wide images while respecting a fixed resolution constraint to simulate the scale-up ability of our pipeline. Experiments were run on an NVIDIA Quadro RTX 6000 with 24GB of RAM. Detailed information about hyperparameters and experiment settings is also provided in the official repository (https://github.com/StefanoSamele-PoliMi/SADSeM).

3.1. Mask-RCNN finetuning procedure and results

In this section, we detail the fine-tuning process for Mask-RCNN. We started from a pre-trained version based on a ResNet50 network. Instead of training different models for each different image size used in the work, we did a joint training of all classes starting from different image resolutions. For each class of each dataset, we have tagged 10 images. The tagging procedure was carried out with a *global-to-local* strategy. We went from the external edges of the object and proceeded to highlight the details. It is a hierarchical strategy that leaves us, at inference time, with two options. Either use the segmentation masks as they are, with possible overlaps, or consider them as a partition of the whole object assigning each pixel of the original image to one class only; we adopted the second choice, specifying an a priori ordering for the sub-classes. If a pixel is identified as belonging to a specific subclass, we do not assign it to any other.

At training time, we randomly sampled a maximum size between (256, 512, 768, 1024) and resized the image such that the maximum dimension coincided with the sampled one, and we extracted a 256x256 crop from it.

The batch size was 10; we adopted stochastic gradient descent (SGD) as the optimizer with a maximum learning rate of 0.001, a momentum equal to 0.9, and a weight decay of 0.0001. We also used a cosine annealing learning rate scheduler [21] down to a minimum learning rate of 0.000001. We also did not freeze the features' extractor network so that the weights could get accustomed to the specific dataset. The rest of the settings were left untouched.

As an example, we have reported in Figs. 5 and 6 the segmentations used for the MVTec AD dataset. This dataset is a benchmark collection for industrial visual anomaly detection and localization, containing over 5000 high-resolution images across 10 object and 5 texture categories: bottle, cable, capsule, hazelnut, metal nut, pill, screw, toothbrush, transistor, zipper, carpet, grid, leather, tile, and wood. We only used object categories for our work. We report in Figs. 7 and 8, qualitative results of the obtained segmentation masks for the classes *cable* and *capsule* of the MVTec AD datasets for different input image resolutions. The segmentation model works perfectly in both cases. We have evaluated the performance of the segmentation model on the MVTec AD dataset test set (only normal images, as we want to assess the performance of MaskRCNN under a standard scenario) and scored a mean Intersection-over-Union (mIoU) of 98.55 %.

3.2. SADSeM ablation studies

We have then systematically analyzed the impact of the different components of our algorithm through ablation studies. We have thus created three variants of our model: the first one does not leverage the segmentation anomaly map score '*No-Segmentation-Masks*', and it uses features coming from Layer 2 and Layer 3, as done in [26]. We are thus simply discriminating the features according to the sub-class they belong to. The second one, named '*Neutral-Backbone (N-B)*', uses the segmentation maps' anomaly score, and the features are extracted from the same layers of a non-fine-tuned network as before. The third one uses features extracted from a fine-tuned network ('*Fine-tuned-Backbone*') but not those assigned through our procedure in Section 2.3. '*Complete*' is the experiment referring to the whole SADSeM pipeline. Last, as a counterproof, we are using the complete pipeline ('*Complete Neutral Backbone*') with features assigned through our procedure but coming from a non-fine-tuned network. This last experiment definitely proves the

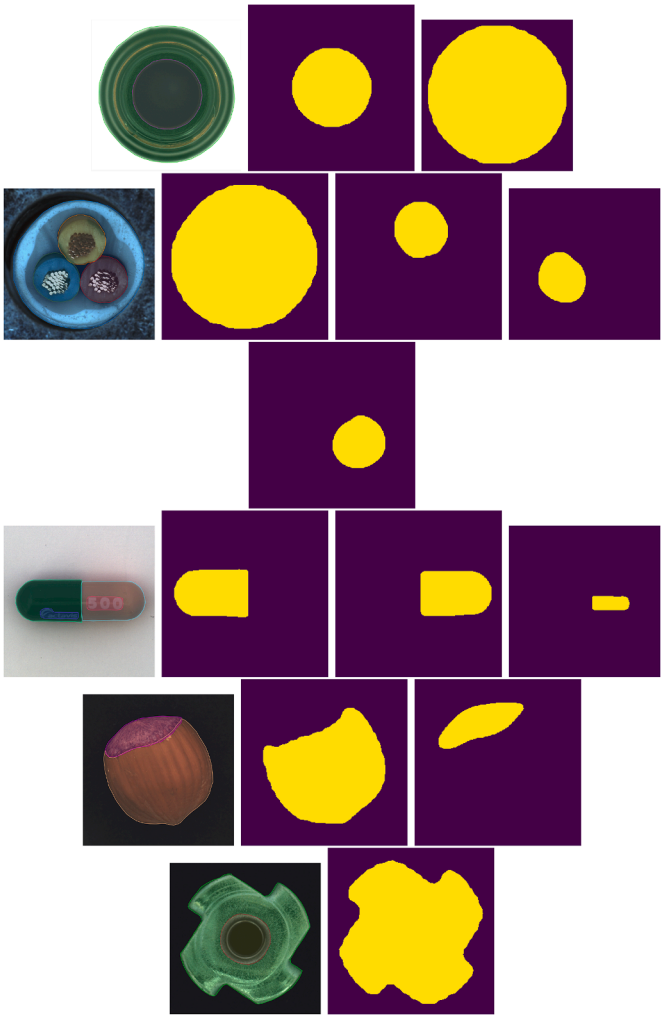

Fig. 5. Examples of the hierarchical tagging procedure adopted for some object classes of MVTec AD dataset (bottle, cable, capsule, hazelnut, metal nut).

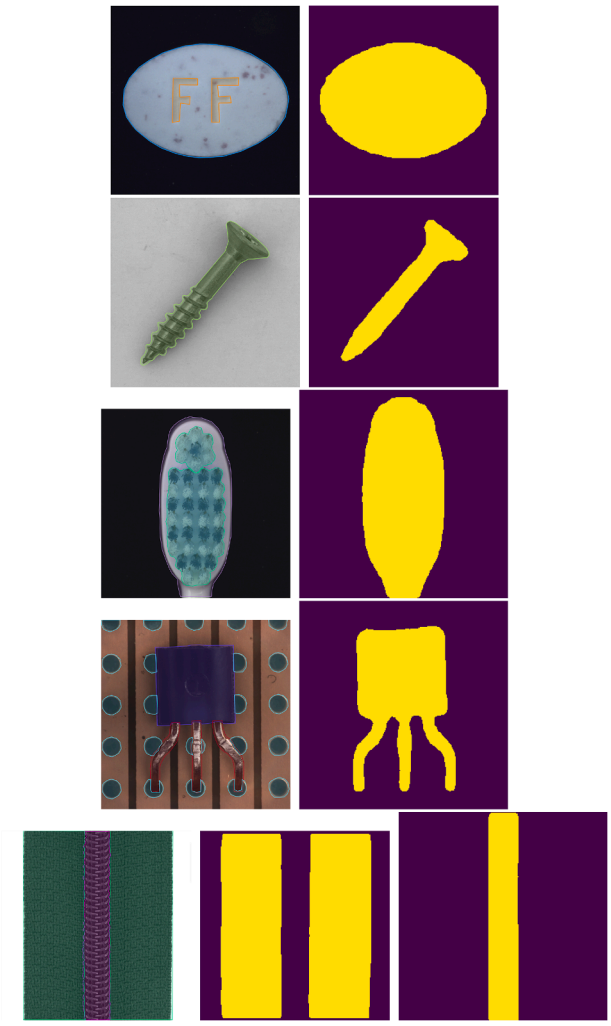

Fig. 6. Examples of the hierarchical tagging procedure adopted for the object classes of MVTec AD dataset (pill, screw, toothbrush, transistor, zipper).

importance of the adoption of the fine-tuned network. All the information about the sub-classes adopted for each class, the corresponding assigned layers, hyperparameters concerning the features resampling procedures, and those related to segmentation masks' anomaly score pipeline are provided in the open-source code.

We run the different models on the MVTec AD dataset. Images are normalized according to ImageNet [12] mean and standard deviation values and resized to 256x256. We measured the performance of our method using the Area Under the Receiver Operating Characteristic curve (AUCROC) at the image and pixel levels. As we can see from Table 1, the '*Complete*' model significantly increases detection and localization scores with respect to the others.

3.3. Comparison with SOTA algorithms

We have compared our method with state-of-the-art anomaly detection methods. We considered three methods published in top-notch con-

Table 2
BTAD dataset Results. The first row shows image-level AUCROC, while the second row shows pixel-level AUCROC. Bold scores highlight the best performance.

	PatchCore [26]	PNI [4]	EfficientAD [5]	SADSeM
Mean	**0.998** **0.982**	0.995 0.979	0.976 0.956	0.996 0.980

Table 3
MVTec AD Dataset Results. The first row shows image-level AUCROC, while the second row shows pixel-level AUCROC. Bold scores highlight the best performance.

	PatchCore [26]	PNI [4]	EfficientAD [5]	SADSeM
Mean	0.987 **0.988**	**0.988** **0.988**	0.960 0.927	0.987 0.976

Table 1
Results of the ablation study on MVTec AD Dataset. The first row shows image-level AUCROC, while the second row shows pixel-level AUCROC. Bold scores highlight the best performance.

	No-Segmentation Masks	N-B	Fine-tuned Backbone	Complete	Complete N-B
Mean	0.964 0.947	0.982 0.975	0.982 **0.976**	**0.987** **0.976**	0.977 0.974

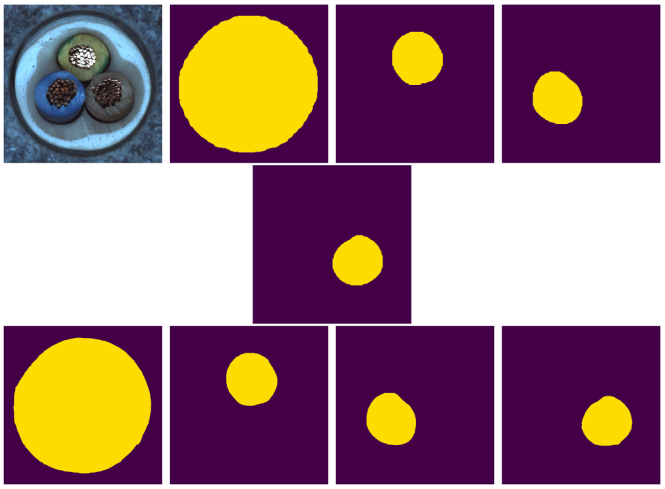

Fig. 7. Original Cable image with segmentation results for a 256x256 (top) and a 1024x1024 (bottom) input image.

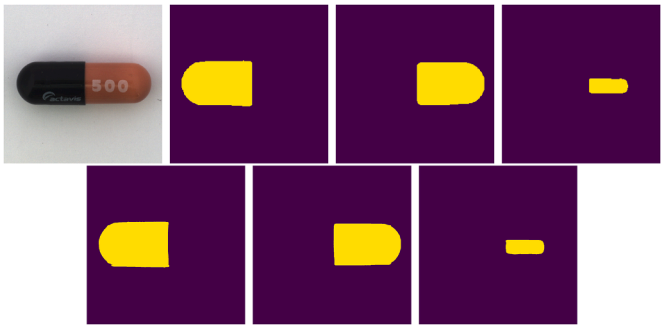

Fig. 8. Original Capsule image with segmentation results for a 256x256 (top) and a 1024x1024 (bottom) input image.

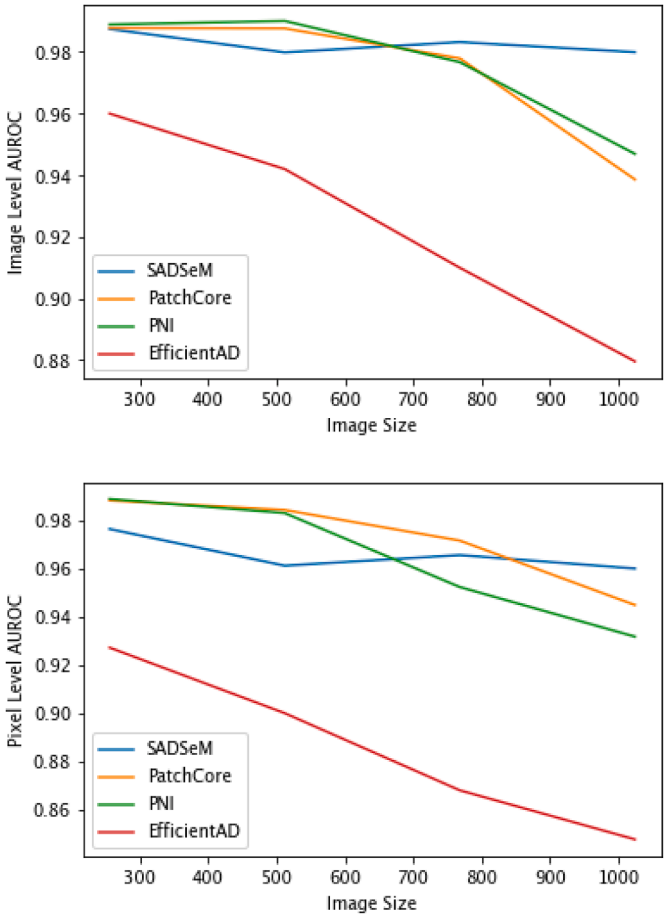

Fig. 9. Performances on MVTec AD dataset measured as image-level AUCROC (top) and pixel-level AUCROC (bottom) as the input image size increases.

Table 4
MVTecLoco AD dataset Results. The first row shows image-level AUCROC, while the second row shows the normalized area under the sPRO curve up to an average false positive rate per pixel of 10 %. Bold scores highlight the best performance.

	PatchCore [26]	PNI [4]	EfficientAD [5]	SADSeM
Mean	**0.844**	0.835	0.786	0.773
	0.646	0.624	0.511	0.538

Table 5
Image-level AUCROC results in the experiment with 1024x1024 pixels input images, on different datasets. Bold scores highlight the best performance.

	PatchCore [26]	PNI [4]	EfficientAD [5]	SADSeM
MVTec AD	0.938	0.946	0.879	**0.979**
MVTec Loco AD	0.724	0.704	0.666	**0.768**
BTAD	0.987	0.987	0.891	**0.995**

ferences with a well-known record track of best performances according to [3]. These methods are: PatchCore [26], (PNI [4]), and EfficientAD [5]. We adopted official implementation when possible. For PNI [4], we did not apply the segmentation refinement network as it is a step that can always be applied to every other method and involves manually hand-crafted data augmentation. All the other settings of the methods are taken directly from the papers and reflect the best possible configuration.

The dataset chosen for this session of experiments are three: MVTec Loco Anomaly Detection dataset (MVTecLoco AD, [8]), MVTec AD dataset, and BeanTech Anomaly Detection dataset (BTAD, [22]). We have chosen these datasets because, from last to first, there is an increase in the complexity of the images and anomalies. The first dataset contains two object classes consisting of simple geometrical shapes with clear patterns; the anomalies interrupt these patterns. The MVTecLoco dataset comprises high-resolution images, ensuring detailed visual representations of objects and anomalies, thereby simulating real-world scenarios more accurately. The dataset has two types of anomalies: structural defects, including irregularities and abnormalities, and logical defects, violating underlying constraints. Again, images are Imagenet-normalized and re-sized to 256x256 for MVTec and BTAD, while for MVTecLoco, we resized the greatest dimension to 256 while the remaining one was calculated to respect the original image ratio. For MVTec and BTAD, we have reported image-level and pixel-level AUCROC (first and second rows of the tables) mean over classes. For MVTecLoco, the nature of logical defects and certain types of structural defects imply that there is no unique possible ground truth segmentation. Thus, the authors suggest the adoption of Saturated Per-Region Overlap (sPRO, [8]). The threshold-independent related metric is the Area Under the

Table 6
Pixel-level AUCROC (except for MVTec Loco AD where AUsPRO with FPR up to 10 % is used) results in the experiment with 1024x1024 pixels input images, on different datasets. Bold scores highlight the best performance.

	PatchCore [26]	PNI [4]	EfficientAD [5]	SADSeM
MVTec AD	0.944	0.931	0.847	**0.960**
MVTec Loco AD	0.473	0.472	0.35	**0.543**
BTAD	0.971	0.973	0.871	**0.975**

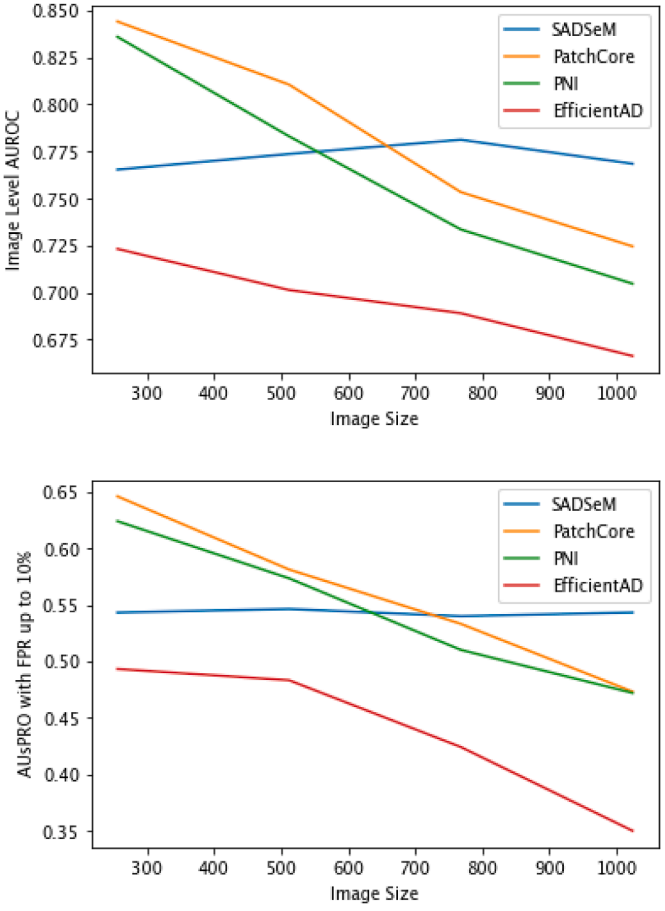

Fig. 10. Performances on MVTecLoco AD dataset measured as image-level AUCROC (top) and pixel-level AUCROC (bottom) as the input image size increases.

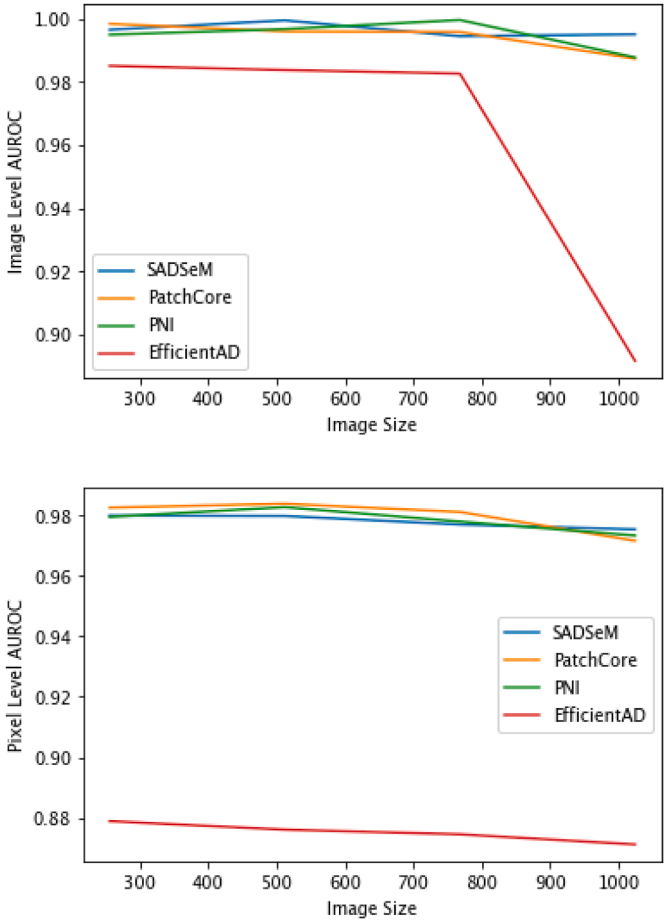

Fig. 11. Performances on BTAD dataset measured as image-level AUCROC (top) and pixel-level AUCROC (bottom) as the input image size increases.

Table 7
Ablation study for crops' stride. The first row shows the average image-level AUCROC, while the second row shows the average pixel-level AUCROC on the MVTec AD dataset. Bold scores highlight the best performance.

Crop Stride	64	128	256
PatchCore [26]	**0.987** 0.984	**0.987** **0.985**	0.986 0.975
PNI [4]	0.988 0.986	**0.990** 0.987	**0.990** **0.988**
EfficientAD [5]	0.925 0.866	**0.941** **0.868**	0.904 0.813

sPRO - False Positive Rate (FPR) curve (up to an average false positive rate per pixel of 10 %.). We adopted this metric instead of the pixel-level AUCROC (always averaging over different classes). We can see from Tables 2–4 that our method achieves good performance, slightly below the state-of-the-art algorithms, all crafted though for that specific image size.

3.4. Scaling capabilities

As stated at the beginning of our work, our interest is to bridge the gap between the theoretical performance of anomaly detection algorithms and real industrial-world scenarios. The proposed method has been developed to achieve good scalability results when dealing with images of larger sizes. Recognizing the object components and their semantic organization is useful when there is no way to gather a unitary view of the object. We assume that we are dealing with an object of several parts that are not enclosed in a small squared image. For example, we can consider the tire of a car. Since its geometrical shape is the torus, its acquisition with a linear camera is a rectangular image, with one dimension way more prominent than the other.

In this testing scenario, we fix an arbitrary maximum image sub-crop size. Models will not be allowed to process sub-crops bigger than this size. The challenge is maintaining consistent performance when dealing with the chosen fixed-size sub-crops still scaling to bigger images. This setting forces the method to be applied several times to different positions of the input image. This means that SOTA methods will have to rely only on the natural alignment of the crops, without other possibilities, to process the image and locate the features. Our method, instead, does not have to look at the whole image to extract features and localize them. We adopted a segmentation model to address work with sub-portions of the whole image. For BTAD and MVTec AD datasets, we have chosen 256 again for crop size, as the models perform well with this setting. The input image will instead scale up to 1024 (256-512-768-1024). For each class of the MVTecLoco AD dataset, instead, we have chosen the biggest dimension starting from 256 and rescaled the other dimension to keep proportions. A key aspect of this last scenario is the possible stride to apply to the crops' extraction. We run a small hyperparameter search on the SOTA methods to identify the best stride value: starting from an input image of 512x512, we extract crops with strides 64, 128, and 256. We can see from Table 7 that for PatchCore [26] and EfficientAD [5], the best scores are achieved when the stride is 128, while a stride of the same size as the image is needed for PNI [4]. For our method, we used non-overlapping sub-crops, as we experimentally verified no changes in performance.

We report in Figs. 9–11 the scores as a function of the increasing input image size for the same datasets and metrics adopted in the previous section. We can clearly see that for realistic products, such as those contained in MVTec AD and MVTec AD Loco, the increased size of the input image drastically lowers the performance of SOTA methods. PNI [4] and PatchCore [26] lose approximately 5 % points for image-level AUCROC and even more at pixel-level Fig. 9. Our method instead, still achieves 0.98 of image-level AUCROC and 0.96 of pixel-level AUCROC.

On MVTecLoco, the drop in image-level AUCROC is even larger: more than 10 % points. Instead, SADSeM can scale the performance, keeping a consistent 0.76 of image-level AUCROC. With almost no oscillation at all, we again overcome the performance of SOTA methods with 1024x1024 inputs. We remark that other methods still perform well on the BTAD dataset due to the fact that the images are made of simple geometric shapes with clear local patterns. Nevertheless, our method shows the expected stability. We have also reported in Tables 5 and 6 the numeric results shown in Figs. 9–11 for the 1024x1024 scenarios.

4. Conclusions

We introduced a novel pipeline that performs Anomaly Detection over high-resolution images of industrial products. We combined fine-tuned segmentation models, feature localization and extraction techniques, and a scoring algorithm. Comparative experiments against state-of-the-art anomaly detection methods on benchmark datasets demonstrated competitive performance. The design of our method prioritizes scalability, ensuring robustness with large image sizes. By tailoring the segmentation stage to handle images of varying resolutions, the method consistently delivered strong results, outperforming existing approaches on high-resolution data.

A current limitation is the need to fine-tune a segmentation model, though in our MaskRCNN setup, this requires as few as 10 annotated images per class. We recognize that manual annotation for supervised segmentation remains labor-intensive, motivating exploration of unsupervised alternatives. Future work will investigate advanced ViT-based segmentation models (e.g., SAM) and zero-shot segmentation methods. While these models are increasingly accessible, adapting them effectively requires more than plug-and-play use: it demands a deep understanding of their latent representations. Future works will also explore the adoption of embeddings coming from these models to understand their properties in the context of industrial anomaly detection.

CRediT authorship contribution statement

Stefano Samele: Conceptualization, Methodology, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Funding acquisition; **Francesco Attorre:** Software, Validation, Investigation, Data curation; **Matteo Matteucci:** Conceptualization, Resources, Supervision, Project administration, Funding acquisition.

Data availability

The code to replicate the paper experiments can be found at https://github.com/StefanoSamele-PoliMi/SADSeM.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Stefano Samele reports equipment, drugs, or supplies was provided by Pirelli Tyre. Francesco Attorre reports equipment, drugs, or supplies was provided by Pirelli Tyre. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Supplementary material

Supplementary material associated with this article can be found in the online version at 10.1016/j.dsp.2025.105613

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] J. An, S. Cho, Variational autoencoder based anomaly detection using reconstruction probability, Spec. Lect. IE 2 (1) (2015) 1–18.
[3] V. Authors, Papers with Code, 2024. https://paperswithcode.com/.
[4] J. Bae, J.-H. Lee, S. Kim, Pni: industrial anomaly detection using position and neighborhood information, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 6373–6383.
[5] K. Batzner, L. Heckler, R. König, Efficientad: accurate visual anomaly detection at millisecond-level latencies, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2024, pp. 128–138.
[6] P. Bergmann, S. Löwe, M. Fauser, D. Sattlegger, C. Steger, Improving unsupervised defect segmentation by applying structural similarity to autoencoders, CoRR, abs/1807.02011, 2018. http://arxiv.org/abs/1807.02011.
[7] P. Bergmann, M. Fauser, D. Sattlegger, C. Steger, Mvtec ad - a comprehensive real-world dataset for unsupervised anomaly detection, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 9584–9592. https://doi.org/10.1109/CVPR.2019.00982
[8] P. Bergmann, K. Batzner, M. Fauser, D. Sattlegger, C. Steger, Beyond dents and scratches: logical constraints in unsupervised anomaly detection and localization, Int. J. Comput. Vis. 130 (4) (2022) 947–969.
[9] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, IEEE Trans. Pattern Anal. Mach. Intell. 40 (4) (2017) 834–848.
[10] T. Defard, A. Setkov, A. Loesch, R. Audigier, Padim: a patch distribution modeling framework for anomaly detection and localization, in: International Conference on Pattern Recognition, Springer, 2021, pp. 475–489.
[11] H. Deng, X. Li, Anomaly detection via reverse distillation from one-class embedding, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 9737–9746.
[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: a large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 248–255.
[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A Large-Scale Hierarchical Image Database, CVPR09, 2009.
[14] T. Ehret, A. Davy, J.-M. Morel, M. Delbracio, Image anomalies: a review and synthesis of detection methods, J. Math. Imaging Vis. 61 (5) (2019) 710–743.
[15] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[16] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2961–2969.
[17] X.-Q. Jiang, K. Huang, S. Zhou, W. Hu, H. Peng, J. Jin, Z. Fang, Dual Flow Reverse Distillation for Unsupervised Anomaly Detection, 2025, p. 105258.
[18] S. Lee, S. Lee, B.C. Song, Cfa: coupled-hypersphere-based feature adaptation for target-oriented anomaly localization, IEEE Access 10 (2022) 78446–78454.
[19] C.-L. Li, K. Sohn, J. Yoon, T. Pfister, Cutpaste: self-supervised learning for anomaly detection and localization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 9664–9674.
[20] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2117–2125.
[21] I. Loshchilov, F. Hutter, SGDR: stochastic gradient descent with warm restarts, in: International Conference on Learning Representations, 2017. https://openreview.net/forum?id=Skq89Scxx.
[22] P. Mishra, R. Verk, D. Fornasier, C. Piciarelli, G.L. Foresti, Vt-adl: a vision transformer network for image anomaly detection and localization, in: 2021 IEEE 30th International Symposium on Industrial Electronics (ISIE), IEEE, 2021, pp. 1–06.
[23] M. Pimentel, D.A. Clifton, L. Clifton, L. Tarassenko, A review of novelty detection, Signal Process. 99 (2014) 215–249.
[24] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, Adv. Neural Inf. Process. Syst. 28 (2015).
[25] D. Rezende, S. Mohamed, Variational inference with normalizing flows, Proceedings of the 32nd International Conference on Machine Learning 37 (PMLR) (2015) 7–09. Proceedings of Machine Learning Research, https://proceedings.mlr.press/v37/rezende15.html.
[26] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, P. Gehler, Towards total recall in industrial anomaly detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 14318–14328.
[27] M. Rudolph, B. Wandt, B. Rosenhahn, Same same but differnet: semi-supervised defect detection with normalizing flows, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2021, pp. 1907–1916.
[28] M. Rudolph, T. Wehrbein, B. Rosenhahn, B. Wandt, Fully convolutional cross-scale-flows for image-based defect detection, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2022, pp. 1088–1097.

[29] M. Rudolph, T. Wehrbein, B. Rosenhahn, B. Wandt, Asymmetric student-teacher networks for industrial anomaly detection, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2023, pp. 2592–2602.
[30] 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.
[31] B. Sun, B. Li, S. Cai, Y. Yuan, C. Zhang, Fsce: few-shot object detection via contrastive proposal encoding, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 7352–7362.
[32] M. Tan, Q. Le, Efficientnet, Rethinking model scaling for convolutional neural networks, in: International Conference on Machine Learning, 2019, pp. 6105–6114.
[33] Y. Tang, L. Zhao, S. Zhang, C. Gong, G. Li, J. Yang, Integrating prediction and reconstruction for anomaly detection, Pattern Recognit. Lett. 129 (2020) 123–130.
[34] J. Wang, W. Huang, S. Wang, P. Dai, Q. Li, Lrgan: visual anomaly detection using gan with locality-preferred recoding, J. Vis. Commun. Image Represent. 79 (2021) 103201.
[35] X. Wang, T.E. Huang, T. Darrell, J.E. Gonzalez, F. Yu, Frustratingly simple few-shot object detection, in: Proceedings of the 37th International Conference on Machine Learning, 2020, pp. 9919–9928.
[36] B.J. Wheeler, H.A. Karimi, A semantically driven self-supervised algorithm for detecting anomalies in image sets, Comput. Vis. Image Understand. 213 (2021) 103279.
[37] J. Wu, S. Liu, D. Huang, Y. Wang, Multi-scale positive sample refinement for few-shot object detection, in Computer Vision-ECCV 2020: 16th European Conference, Glasgow, UK, Springer, 2020. 456–472. Proceedings, Part XVI 16.
[38] X. Xie, Z. Li, S. Xiong, Z. Liu, T. Cai, Memflow-ad: an anomaly detection and localization model based on memory module and normalizing flow, J. Vis. Commun. Image Represent. 110 (2025) 104454.
[39] H. Xu, S. Xu, W. Yang, Unsupervised industrial anomaly detection with diffusion models, J. Vis. Commun. Image Represent. 97 (2023) 103983.
[40] M. Yang, P. Wu, H. Feng, Memseg: a semi-supervised method for image surface defect detection using differences and commonalities, Eng. Appl. Artif. Intell. 119 (2023) 105835.
[41] V. Zavrtanik, M. Kristan, D. Skocaj, Reconstruction by inpainting for visual anomaly detection, Pattern Recognit. 112 (2021) 107706. https://doi.org/10.1016/j.patcog.2020.107706
[42] V. Zavrtanik, M. Kristan, D. Skočaj, Draem - a discriminatively trained reconstruction embedding for surface anomaly detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 8330–8339.