Divide and Conquer: High-Resolution Industrial Anomaly Detection via Memory Efficient Tiled Ensemble

Blaž Rolih[1] Dick Ameln[2] Ashwin Vaidya[2] Samet Akcay[2]
[1]University of Ljubljana, Faculty of Computer and Information Science
[2]Intel
br9136@student.uni-lj.si, {dick.ameln, ashwin.vaidya, samet.akcay}@intel.com

Abstract

Industrial anomaly detection is an important task within computer vision with a wide range of practical use cases. The small size of anomalous regions in many real-world datasets necessitates processing the images at a high resolution. This frequently poses significant challenges concerning memory consumption during the model training and inference stages, leaving some existing methods impractical for widespread adoption. To overcome this challenge, we present the tiled ensemble approach, which reduces memory consumption by dividing the input images into a grid of tiles and training a dedicated model for each tile location. The tiled ensemble is compatible with any existing anomaly detection model without the need for any modification of the underlying architecture. By introducing overlapping tiles, we utilize the benefits of traditional stacking ensembles, leading to further improvements in anomaly detection capabilities beyond high resolution alone. We perform a comprehensive analysis using diverse underlying architectures, including Padim, PatchCore, FastFlow, and Reverse Distillation, on two standard anomaly detection datasets: MVTec and VisA. Our method demonstrates a notable improvement across setups while remaining within GPU memory constraints, consuming only as much GPU memory as a single model needs to process a single tile. [1] [2]

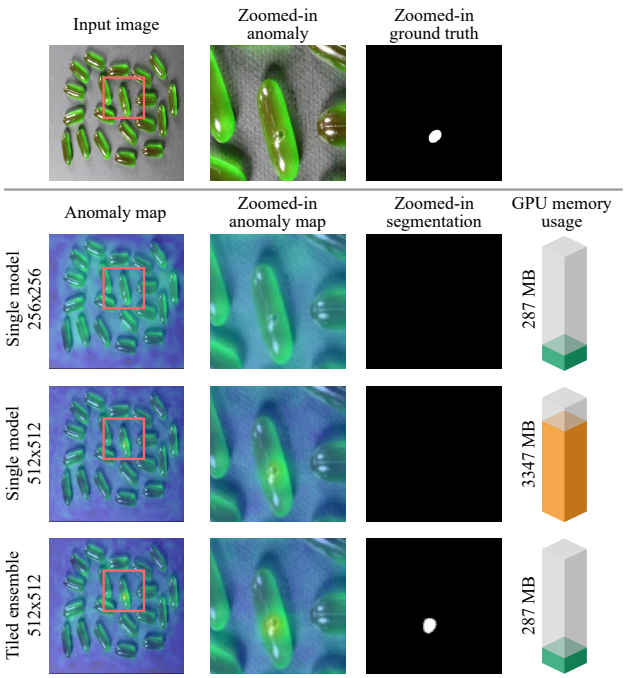

Figure 1. Example of anomaly localization on VisA capsules category. A tiled ensemble successfully manages to detect an anomaly, while a single model with a smaller resolution or equivalent resolution fails to do so. The tiled ensemble achieves this while remaining within the GPU memory constraints.

1. Introduction

The detection and localization of anomalies in images is a crucial task with a wide range of industrial applications. The ability to identify hard-to-detect defects of various sizes within images enables automation of many processes, maintenance of safety, and prevention of financial loss.

In recent years, the field has witnessed substantial performance improvements, primarily driven by advancements in deep learning techniques. In addition, the high real-world potential of anomaly detection techniques has prompted a shift of focus towards efficiency, as latency, throughput, and memory consumption are important metrics to optimize when deploying models on resource-limited devices [4, 16, 22, 24, 26, 40].

A challenge of real-world datasets is that the size of the anomalous regions within the images may be very small relative to the full image size. The common practice of down-

[1]Available as part of Anomalib: https://github.com/openvinotoolkit/anomalib.
[2]Research conducted during GSoC 2023 at OpenVINO.

scaling the input images to a predetermined input size may in such cases lead to a loss of pixel-level information, which in turn causes the model to miss small anomalies and incorrectly mark images as defect-free. Processing the images at their original resolution or reducing the amount of downscaling may constitute a natural tactic to prevent this type of false negative, but will at the same time inflate the memory consumption of the model. As a result, memory constraints may prevent processing the images in a resolution suitable for detecting the smallest anomalies in the dataset, especially in low-resource settings.

Tiling mechanisms, which subdivide the input images into a rectangular grid of tiles as a pre-processing step, have been used to process images at a high resolution while keeping memory use low [25]. By passing individual tiles to the model as input instead of full images, tiling reduces the model's input dimensions, while maintaining the effective input resolution of the images content-wise. This approach may not always be ideal, particularly for methods sensitive to object alignment [23], as using a single model for all patches may compromise spatial information preservation.

Contrary to traditional tiling, our tiled ensemble trains a separate model on each of the individual tile locations. The full training procedure yields an ensemble of independently trained models, each specialized on a single specific tile location. By assigning a separate model to each tile location, we achieve a direct spatial mapping of feature space to pixel space, making the method suitable for spatially-aware models. An additional advantage of using separate models is that it allows us to leverage the benefits of stacking ensemble methods by introducing overlapping tiles, which further improves anomaly detection performance. By merging the predictions of the individual models as a post-processing step, we obtain an end-to-end pipeline, offering direct application to practical settings while ensuring that the peak GPU memory usage remains in the range of that required by a model processing a single tile. Since our approach only changes the pre- and post-processing stages, it is not limited to a specific model architecture and can be applied as an extension to any anomaly detection pipeline. Figure 1 illustrates how a tiled ensemble can detect small anomalies by utilizing increased resolution without consuming excessive GPU memory.

To showcase the applicability of our tiled ensemble, we benchmark the approach against non-tiling baselines, as well as traditional tiling methods, using a diverse set of architectures such as probability density modelling (Padim [11]), memory bank based (Patchcore [29]), student-teacher (Reverse Distillation [12]), and normalizing flows (Fastflow [40]). For evaluation, we use two well-known anomaly detection datasets: MVTec AD [6] and VisA [46], with an emphasis on detecting smaller anomalies, particularly evident in the VisA dataset.

Overall, this paper provides the following contributions:

- We propose a practical approach for the detection and localization of anomalies in high-resolution images while adhering to GPU memory constraints. This enables the detection of small anomalies in real-world applications, increasing reliability and performance.
- Our approach offers a model-agnostic framework that can enhance both existing and upcoming anomaly detection architectures. By adopting our methodology, these architectures can better handle higher-resolution images, additionally benefiting in constrained settings without the need for modification of the underlying architecture.
- Having a dedicated model for each tile location enables the model to highly specialize in specific part of an image. Additionally, the integration of overlapping tiles in our approach has the advantages of conventional stacking ensembles. This results in enhanced performance that surpasses what can be achieved solely through increased resolution.

2. Related work

In recent times, there have been notable advancements in the field of visual anomaly detection, with numerous techniques being introduced based on various approaches such as reconstructive methods [5, 8, 43], student-teacher networks [4, 7, 12, 31, 36], discriminative methods [14, 23, 38, 42, 44, 45], normalizing flows [16, 30, 40], and embedding-based methods [11, 24, 29].

Apart from the design of novel model architectures, an active area of focus has been enhancing and extending existing approaches. Recent research has indicated that the performance of anomaly detection models can benefit from careful design choices around data augmentation[37], pre-training characteristics[17], and feature space selection [19]. Other studies have focused on modifying or extending the architecture of existing models. Ristea et al. [28] introduced the SSPCAB block, which can be injected into various state-of-the-art methods to enhance their performance. Similarly, e Silva et al. [13] demonstrated that significant improvements can be achieved by extending existing architectures with attention blocks. Finally, Heckler. and König. [18] developed a feature selection method that optimally selects a layer from the pre-trained feature extractor depending on the characteristics of the task. The tiled ensemble method follows a similar strategy, altering the pre- and post-processing stages of existing anomaly detection pipelines with the aim of improving the performance on datasets with small anomalies.

A common class of anomaly detection models is patch-based models. A patch refers to a spatial location in the intermediate feature embedding map of the model backbone and usually translates directly to a pixel area in the input images. Patch-based models aim to find the natural

distribution of each patch location from the feature embeddings of normal images during training and estimate the distance of the feature embeddings to this distribution during inference. This process yields a set of patch-level anomaly scores which form the basis of the anomaly localization predictions. To find the distribution of a given patch location, a model may rely only on the embeddings of that same patch location [10, 11, 33, 39], or alternatively also consider the interrelation among patches [26, 29, 35]. Further, the authors of CutPaste [23] discovered that employing separate models for each patch location yields superior results. This insight forms the basis of assigning a dedicated model for each tile location in our tiled ensemble method, which further extends this into a generic extension for any anomaly detection architecture.

The reduction of memory use in anomaly detection is a common research topic [4, 16, 22]. This is especially relevant for fields such as pathology [34], where a high input resolution is needed to distinguish the anomalous characteristics within the images [25]. Processing images at a higher resolution prevents missing small anomalies but at the same time leads to increased GPU memory usage. In light of these challenges, the tiled ensemble extends the capabilities of various anomaly detection methods to enable efficient processing of high-resolution images. This approach capitalizes on findings from Heckler et al. [19], which explores how image resolution impacts the performance of anomaly detection architectures.

The ensemble approach is often used to increase the performance of a base model. Several individual models are combined, which results in a better generalization performance [15], accuracy, stability, and reproducibility [9]. Ensembles are increasingly popular in visual anomaly detection and have shown promising results [20, 27, 32, 41]. Bergmann et al. [7] used an ensemble of students to mimic the teacher network. Recent anomaly detection methods employ ensembles by keeping the architecture consistent while using a different backbone for feature extraction [3, 21, 29]. In each of these studies ensemble models consistently outperform single models, achieving state-of-the-art results in anomaly detection.

3. Method

The tiled ensemble method is structured as a series of sequential steps. The method initially divides the high-resolution image into tiles (Section 3.1), followed by training individual models for each tile location (Section 3.2). Once predictions are obtained, a merging mechanism is utilized to produce full-image-level data (Section 3.3), followed by standard post-processing steps.

A high-level overview of the workflow is presented in Figure 2. The approach encapsulates all the training and inference steps in a pipeline, which enables immediate application for industrial use cases that involve the analysis of high-resolution images.

3.1. Tiling

To reduce the memory footprint of a high-resolution image, the first step is to split the image into a set of tiles, which are then separately processed by an individual model. For an image $X \in \mathbb{R}^{c \times h \times w}$, tile size $h^t \times w^t$ and stride s_h, s_w, this set of tiles $\mathcal{T}$ is defined as

$$\begin{aligned} \mathcal{T} = \{T_{i,j} \in \mathbb{R}^{c \times h^t \times w^t} | \\ i \in [0, ..., \left\lfloor \frac{h - h^t}{s_h} \right\rfloor], h^t \leq h, s_h \leq h \\ j \in [0, ..., \left\lfloor \frac{w - w^t}{s_w} \right\rfloor], w^t \leq w, s_w \leq w \\ h^t, w^t, s_h, s_w \in \mathbb{N}\} \end{aligned} \tag{1}$$

If the stride and tile size don't precisely match the image, the image is padded with zeros, which are later removed during untiling. Each tile spans the following pixels of the original image:

$$\begin{aligned} T_{i,j} = \{(a, b) | a \in [s_h * i, ..., s_h * i + h^t), \\ b \in [s_w * i, ..., s_w * i + w^t)\} \end{aligned} \tag{2}$$

The pixels that tiles cover can also overlap in case of stride smaller than tile size, i.e. $s_h < h^t$ or $s_w < w^t$.

For instance, consider an input image with dimensions 512×512 (height $h = 512$ and width $w = 512$), a tile size of 256×256 (tile height $h^t = 256$ and tile width $w^t = 256$) and a stride of $s_h = 128$ and $s_w = 128$. This configuration yields 9 overlapping tiles, labeled $T_{0,0}$ through $T_{2,2}$. The overlapping area between tiles $T_{0,0}$ and $T_{0,1}$ consists of pixels $(a, b) | a \in [0, ..., 256) \wedge b \in [128, ..., 256)$, which means it encompasses the right half of $T_{0,0}$ and the left half of $T_{0,1}$.

Tiling the input image enables predicting images with higher resolution while allowing models to be trained on smaller inputs. This approach significantly reduces GPU memory consumption. Another benefit of tiling is that each model is only responsible for the designated tile location. This localized processing ensures that anomalies detected within a specific tile do not influence the detection results in adjacent or distant tiles, which, overall, prevents the trigger of unrelated spurious predictions across the image.

3.2. Training and inference

Training. By splitting the image into smaller tiles, the problem of high GPU memory consumption is efficiently addressed. However, training a single model on the combined set of all tile locations could lead to a loss of positional information, with a potential negative effect on the

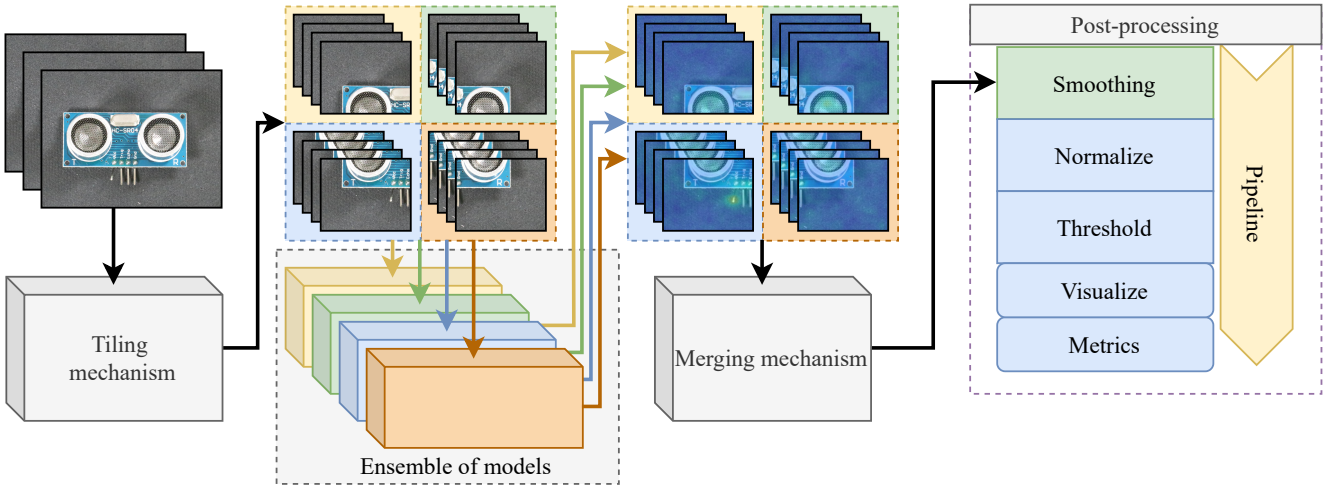

Figure 2. High-level tiled ensemble workflow: Images are first divided into tiles, and separate models are trained for each tile location. Predictions are generated individually for each tile, merged back together, and finally post-processed. Note that tiles can also be overlapping, yet the training is independent for each model.

performance of models that perform better on aligned objects. The tiled ensemble approach addresses this by employing a separate model for each tile location, where the underlying model architecture of the individual models remains unchanged.

Formally, a separate model $M_{i,j}$ is trained for each tile location in the entire set of tiles $\mathcal{T}$, defined in Equation (1), resulting in a set of models:

$$\mathcal{M} = \{M_{i,j} | M_{i,j} \text{ trained on } T_{i,j}\} \tag{3}$$

The tiled ensemble method requires no further modifications to the training process, which is similar to that of a single model. Since each tile location is processed independently, it is possible to train the models in parallel across different devices.

Inference. Once all the models for all locations are trained, the same tiling procedure is followed and each tile is processed by the corresponding model in inference. For tile $T_{i,j}^{test}$ in inference time and model $M_{i,j}$, the pixel-level anomaly map $\mathcal{A}_{i,j}$ and anomaly score $s_{i,j}$ is obtained as:

$$\mathcal{A}_{i,j}, s_{i,j} = M_{i,j}(T_{i,j}^{test}) \tag{4}$$

In this case, the score $s_{i,j}$ is obtained as specified by the underlying architecture. This can either be achieved as a separate process or by taking the maximum value from $\mathcal{A}_{i,j}$.

Due to the independence of predictions, the storage of each tile predictions can also be efficiently managed. By moving the tile predictions to the main memory, the GPU memory usage remains within the constraints.

3.3. Merging

A merging mechanism is utilized to produce a full-resolution anomaly map $\mathcal{A}$ and the score s from individual tile predictions $\mathcal{A}_{i,j}$ and $s_{i,j}$ (Figure 3).

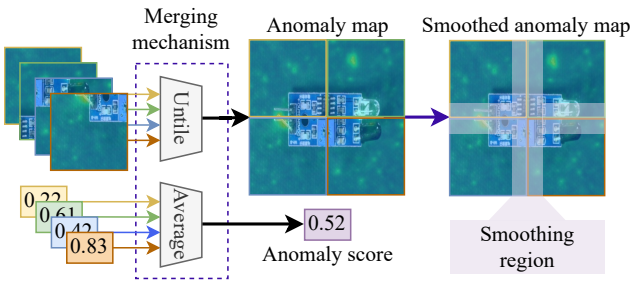

Figure 3. Overview of merging procedure and smoothing. Tile-level anomaly maps are untiled into a full image, with pixel-wise averaging applied to overlapping regions. Anomaly scores from all tiles are averaged, producing a single image-level score. After the predictions are merged, the first step of post-processing involves smoothing the region around the tile seams to enhance the quality of the anomaly map.

Tile-level anomaly maps $\mathcal{A}_{i,j}$ are simply untiled back into a full-image representation to get $\mathcal{A}$. As the tiles can also be overlapping, a pixel-wise averaging strategy [1] is applied on overlapping regions.

Different strategies can be used to tackle the merging of image-level scores $s_{i,j}$. An image can be classified as anomalous as soon as one of the patches is anomalous [29]. Alternatively, the score over all the tiles can be averaged to obtain a single score, which is in our case the default option:

$$s = \frac{1}{N} \sum_{i,j} s_{i,j} \tag{5}$$

The borders of the tiles create seams, leading to undesirable disturbances in the image. To mitigate this issue and enhance the outcomes, a Gaussian filter is applied for smooth-

ing. This smoothing process is confined to a narrow region surrounding the seam, as depicted in Figure 3. The default width of the smoothing region is 10% of the tile width on each side of the seam. In line with standard anomaly detection procedures, the final classification and localization predictions can be obtained by applying a thresholding mechanism to the image-level anomaly scores and anomaly maps respectively.

4. Experiments

An in-depth analysis of the tiled ensemble across multiple configurations and anomaly detection architectures is conducted. The following sections outline the protocols and setups employed to assess the impact of the tiled ensemble.

4.1. Experimental details

Datasets. The method is evaluated on two established industrial datasets: MVTec AD [6] and VisA [46]. MVTec AD and VisA datasets comprise 15 and 12 categories, respectively. Each category consists of a training set containing only normal images and a test set, comprising both normal and anomalous images, with their corresponding pixel-precise ground truth annotations. The anomalies vary in types, shapes, and scales, with the prevalence of larger anomalies in MVTec AD and smaller defects in VisA. An analysis of defect scales is presented in Appendix A.

For both datasets, the images are of high resolution and are resized according to the specified dimensions in the experimental setups, as detailed in the following sections.

Evaluation Metrics Both image and pixel-level performance are evaluated using standard anomaly detection metrics. For image-level anomaly detection, the Area Under the Receiver Operator Curve (AUROC) is employed. To evaluate pixel-wise performance in anomaly localization, the Area Under the Per-Region-Overlap Curve (AUPRO) is used.

The exact protocol outlined by the authors of EfficientAD [4] is followed to obtain latency, throughput, and inference GPU memory consumption. The only exception is the usage of a batch size of 8 instead of 16 for PatchCore, due to excessive GPU memory usage in the case of a single model with 512×512 resolution. For a tiled ensemble, the benchmark inference step encapsulates tiling, inference on all tiles, and untiling. Experiments were conducted on a system with Intel(R) Xeon(R) Gold 5320 CPU and Nvidia Tesla A100 GPU (training) and Nvidia Tesla V100S (inference).

4.2. Evaluation setups

To comprehensively evaluate and compare the performance of the tiled ensemble, four architectures from diverse paradigms are employed. Padim [11] covers probability density modelling, Patchcore [29] is a memory bank based approach, Reverse Distillation [12] represents student-teacher architectures, and Fastflow [40] normalizing flows. Each architecture is then trained in six different setups, where two use a single model with varying resolution, two employ tiled input to a single model and the remaining two utilize the tiled ensemble.

Single model with 256px image size – *SM256*. A single model for each architecture is trained with an input size of 256×256 pixels, aligning with the tile size of the ensemble models. While the effective final resolution processed by this setup is smaller, this setup serves as a baseline as this resolution is the most common in other works.

Single model with 512px image size – *SM512*. To explore the effect of resolution without ensembling, a single model is trained with an input image size of 512×512 pixels. In this case, the model processes the same effective resolution as our base tiled ensemble. However, it consumes a larger amount of GPU memory. This setup allows for a comparison of memory usage and the extent to which the benefits result from ensembling rather than increased resolution.

Tiled ensemble with 9 overlapping 256px tiles – *ENS9*. The base tiled ensemble setup has image resolution of 512×512 pixels, which is then divided into 9 overlapping 256×256 tiles ($h^t = w^t = 256$, $s_h = s_w = 128$). The final predicted anomaly map maintains the same dimensions as the input image, i.e. 512×512. This setup is utilized to highlight the effects of ensembling properties in addition to the ability to process high resolution while adhering to memory constraints.

Tiled ensemble with 4 non-overlapping 256px tiles – *ENS4*. In this setup, the input image has a resolution of 512×512 and is divided into four non-overlapping 256×256 tiles ($h^t = w^t = 256$, $s_h = s_w = 256$). The dimension of the predicted anomaly map remains consistent with the input image, i.e. 512×512s. This setup is utilized to highlight the efficient processing of high resolution, without the additional benefits of multiple (overlapping) predictions as in *ENS9*.

Single model with 9 overlapping 256px tiles – *ST9*. This setup involves using a single model trained on tiled input. The image resolution remains at 512×512 pixels, divided into nine overlapping 256×256 tiles ($h^t = w^t = 256$, $s_h = s_w = 128$), and stacked batch-wise. A single model in this case receives a 256×256 tile for input. This setup is used to compare the effect of having a separate model in a tiled ensemble specializing solely in a single tile location.

Single model with 512px with 4 non-overlapping 256px tiles – *ST4*. Matching the tiled ensemble setup without overlapping tiles, this setup explores how tiling the input works in the case of utilizing a single model for all tile locations. Here, a 512×512 input image is split into

four non-overlapping 256×256 tiles ($h^t = w^t = 256$, $s_h = s_w = 256$), and stacked batch-wise. A single model is then trained on these tiles, with an input size of 256×256 pixels.

Common properties. Each setup is trained on every category, with every run repeated 3 times using a different random seed. A consistent batch size of 32 is used, except for Patchcore where a batch size of 8 is used due to memory limitations. The backbone used in all setups is ResNet18, to keep the comparison fair. Following [4, 19] FastFlow, and Reverse Distillation are limited to 200 steps for all setups. Other properties are kept the same as provided by the original authors of the models and as implemented in Anomalib [2].

5. Results

Results on MVTec AD. Table 1 reports anomaly detection and localization results obtained on the MVTec AD dataset. A tiled ensemble with overlapping tiles (ENS9) achieves the best results in terms of anomaly detection for Padim and FastFlow and second best for PatchCore and Reverse Distillation. It also achieves the best localization performance for PatchCore and FastFlow, and second best for Padim.

Setup	PatchCore	Padim	FastFlow	Reverse Distillation
SM256	97.7/92.8	89.2 / **91.2**	93.1 /89.1	**90.8** / **89.5**
SM512	**98.0** / 94.5	83.0/ 91.0	90.5/88.5	78.5/ 87.2
ST4	97.8 /94.0	83.8/90.6	91.4/85.0	80.7/86.2
ST9	97.8 /94.3	83.3/90.6	90.1/82.8	77.3/ **89.5**
ENS4	96.5/94.1	87.3/90.5	91.8/ 89.5	84.5/82.6
ENS9	97.8 / **95.3**	**89.7** / 91.0	**95.0** / **91.4**	87.8 /82.6

Table 1. Results in anomaly detection and localization (AUROC/AUPRO) on MVTec AD. Best and second best results are marked. A mean of 3 runs is reported for each setup.

Results on VisA with all 6 setups are displayed in Table 2. A tiled ensemble with overlapping tiles (ENS9) achieves the best anomaly detection results for Padim, FastFlow, and Reverse Distillation, significantly outperforming baseline single model (SM256) and single model processing the same resolution of 512×512 (SM512) in all three cases. ENS9 also achieves the best anomaly localization results for FastFlow and Reverse Distillation.

The anomaly detection results of a tiled ensemble (ENS4 and ENS9) consistently outperform a single model with tiled input (ST4 and ST9) in almost all setups, except for PatchCore, and in terms of localization for Padim. This indicates that having a separate model specialized in each tile location can lead to better performance in high-resolution images. The tiled ensemble with overlapping tiles (ENS9) in most setups outperforms a single model processing the same resolution (SM512) as well as a tiled ensemble with non-overlapping tiles (ENS4). This demonstrates the potential benefits of the stacking ensemble mechanism.

Setup	PatchCore	Padim	FastFlow	Reverse Distillation
SM256	92.0/87.7	83.7/82.1	87.4/81.4	83.2/86.9
SM512	**97.2** / **94.0**	81.9/86.9	89.8 / 88.4	71.5/89.3
ST4	95.2/93.6	81.8/ **87.2**	87.1/83.4	72.5/89.3
ST9	96.7 / 93.7	82.3/ 87.1	85.6/78.6	80.6/88.9
ENS4	93.1/93.0	83.8 /86.9	89.4/86.8	88.3 / 89.6
ENS9	95.4/ 93.7	**86.3** /86.9	**92.5** / **89.2**	**91.4** / **90.0**

Table 2. Results in anomaly detection and localization (AUROC/AUPRO) on VisA. Best and second best results are marked. A mean of 3 runs is reported for each setup.

In the tiled ensemble setup, the kNN search in PatchCore's memory bank is limited to embeddings from within the same tile location, whereas single-model setups provide access to embeddings from the entire image. This may explain why PatchCore tends to benefit from a single-model setup. In both MVTec AD and VisA, Reverse Distillation and Padim struggle if the resolution is increased without utilizing the tiled ensemble, showing subpar performance when comparing SM512 to baseline SM256. In the case of MVTec AD, Reverse Distillation still works best with the baseline model, indicating that for some architectures and large anomalies, a tiled ensemble is not necessarily needed.

More detailed results for each category and setup with included standard deviation on MVTec AD and VisA are included in Appendix D.

GPU memory usage. Inference GPU memory and training GPU memory usage are presented in Figure 5, respectively. The tiled ensemble is unaffected by the number of tiles as long as the tiles have the same resolution. During inference, memory consumption remains comparable to that of a single model processing the resolution equivalent to a tile (*Single model 256*), across all models.

Inference GPU memory consumption holds significant importance for end-applications, but training memory consumption also poses a challenge with larger image resolutions. The tiled ensemble maintains a manageable memory footprint in both training and inference, roughly equating to the memory consumption of a single model (Figure 5). Note that the relative memory advantage of the tiled ensemble further increases for higher effective image resolutions (provided that the tile size remains the same), as the memory consumption of each individual model is only related to the tile size.

Effect of resolution on performance and GPU memory usage. A case study is performed on the PCB3 category of

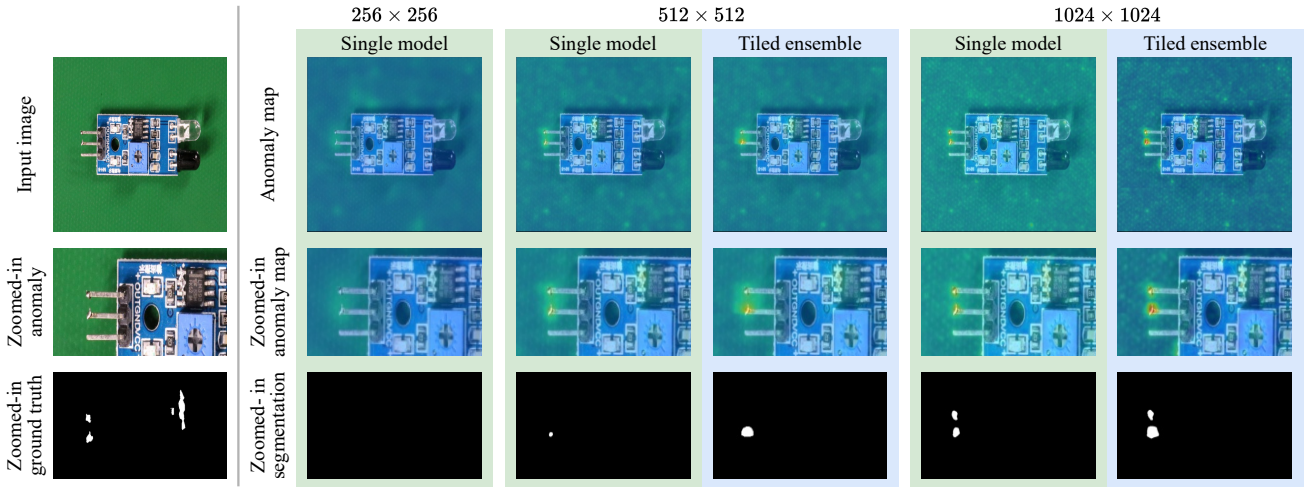

Figure 4. Effects of increasing resolution for Padim on VisA PCB3 category. The first column contains full and zoomed-in input images with corresponding ground truth. The next columns depict full and zoomed-in anomaly maps with their corresponding binary segmentation. The resolution of results is written above each block. Notice how localization improves with higher resolution. The tiled ensemble uses a setup with $h^t = w^t = 256$, $s_h = s_w = 256$.

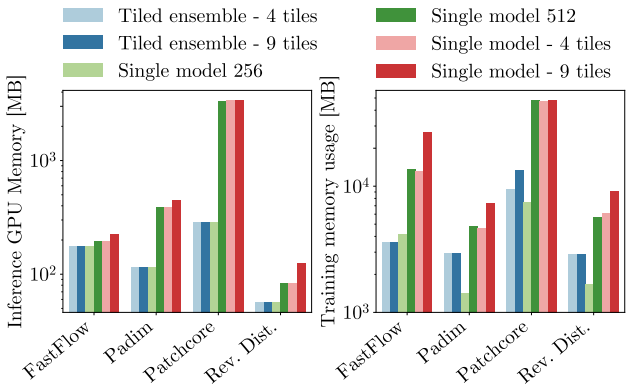

Figure 5. Inference and training GPU memory usage. The memory consumption of the tiled ensemble remains within the range of a model processing an image with the resolution of a single tile (*Single model 256*). This for some models results in a notable reduction, particularly evident in models such as Patchcore.

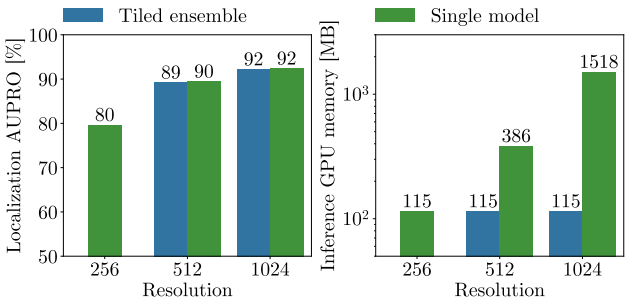

Figure 6. Localization results in terms of AUPRO on VisA PCB3 category for a single model and tiled ensemble with different resolutions. The corresponding memory consumption of each setup is shown on the right. The tiled ensemble uses a setup with $h^t = w^t = 256$, $s_h = s_w = 256$.

the VisA dataset, which contains many small defects that can benefit from increased resolution. The Padim model is used to explore memory consumption and verify the effect of resolution on localization performance. A tile size of 256 with stride 256 is used for ensemble setup ($h^t = w^t = 256$, $s_h = s_w = 256$). The results of localization performance with respect to resolution are presented in Figure 6 with GPU memory consumption also reported for each setup.

Increased resolution offers increased localization performance, most notably showing an improvement in the initial increase from 256×256 to 512×512, at which many small anomalies already become better detectable. While the memory of a single model processing a larger resolution steeply increases, the tiled ensemble maintains the same memory consumption for all resolutions. Fig. 4 contains a qualitative example depicting the localization of a small anomaly from this experiment.

Effect of input resolution on small anomaly detection. Figure 7 illustrates how small anomaly detection may benefit from the higher input resolutions that can be achieved by the tiled ensemble approach. Compared to the 256×256 baseline, the tiled ensemble achieves a notable boost in both detection and localization performance for datasets in which the average size of the anomalous regions is small. As the average size of the anomalies increases, the effect diminishes and the performance of both setups converges. By increasing the effective input resolution, the tiled ensemble

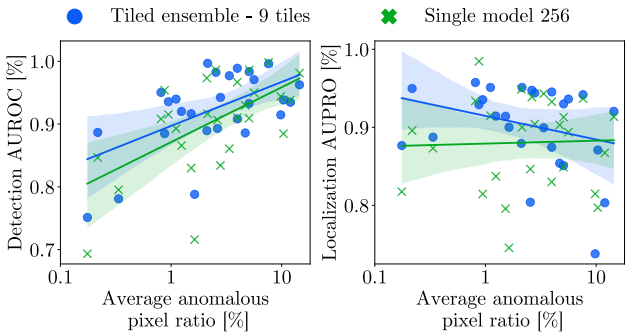

Figure 7. Effect of defect size on anomaly detection and localization performance of the tiled ensemble (ENS9) and single model with 256×256 resolution (SM256). Each point represents a single dataset category from MVTec AD or VisA. Trend lines and confidence intervals added for interpretability. X-axis: average number of anomalous pixels per defective image relative to image size. Y-axis: average performance of setup across all four model architectures.

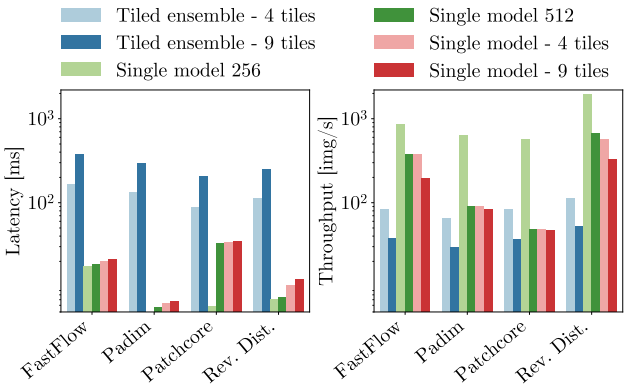

Figure 8. Latency and throughput measured for each setup on an Nvidia Tesla V100S. While the latency considerably increases, throughput sees a relatively smaller reduction for tiled ensemble configurations.

approach facilitates the detection and localization of small anomalous regions that would otherwise go unnoticed as a result of downscaling the input images.

Latency and throughput. The latency and throughput are presented in Figure 8. The low throughput and high latency of ENS4 and ENS9 can be attributed to the increased computational complexity of these setups (Appendix B) and show that the performance advantage of the tiled ensemble comes at the cost of an increased runtime. The latency overhead likely stems from the time needed to transfer individual models to GPU and back, which does not affect throughput as significantly since the model's time on GPU is better utilized. For some models like Patchcore, the throughput of a 4-tiled ensemble exceeds that of a single model processing an equivalent resolution. The preliminary studies showed that the time needed for tiled ensemble inference on GPU still outperforms the inference of a single model with increased resolution on CPU in terms of latency by around 4 times, and in terms of throughput by around 80 times. The training time of all setups is presented in Appendix C.

6. Conclusion

This paper introduces a tiled ensemble approach to effectively detect and localize small anomalies in high-resolution images, which has been a challenge due to the high GPU memory demands required by existing approaches. The tiled ensemble approach addresses this by dividing the image into smaller tiles and training a dedicated model for each tile location. This strategy ensures that the GPU memory usage remains comparable to that of a single model that processes an image the size of one tile. By employing overlapping tiles, the tiled ensemble also takes advantage of the performance improvements associated with traditional stacking ensemble methods, which further improve performance compared to those achievable by simply increasing image resolution.

The tiled ensemble is designed to be easily integrated into current anomaly detection architectures without necessitating any architectural changes, which makes it a flexible and practical solution for small anomaly detection within high-resolution imagery. In an extensive evaluation using various model architectures and two established datasets, the method demonstrated notable performance improvement compared to setups processing images at a lower resolution or without employing a tiled ensemble, with a particularly pronounced impact on datasets with small anomalies.

The results presented in this paper demonstrate the feasibility of applying existing or next-generation anomaly detection models within high-resolution imagery, which opens up new possibilities across various industries.

Limitations. Despite its promising statistical results, our approach has a notable latency overhead, which can be partially mitigated through batched inference. This can be a reasonable sacrifice in cases where the resolution is very large, to enable detection with resolutions that previously were not feasible. As suggested by Heckler et al. [19] and verified by Heckler. and König. [18], strategically choosing a single layer can outperform an ensemble of multiple layers and backbones in certain scenarios. Building on this insight, future research should investigate whether selecting the most suitable layer, or set of layers, for each tile location could further optimize anomaly detection in high-resolution images. Finally, the experiments of the current study did not cover logical anomaly detection benchmarks, which could potentially suffer from a loss of global context as a result of processing each tile location separately.

References

[1] Philip A Adey, Samet Akçay, Magnus JR Bordewich, and Toby P Breckon. Autoencoders Without Reconstruction for Textural Anomaly Detection. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021. 4

[2] Samet Akcay, Dick Ameln, Ashwin Vaidya, Barath Lakshmanan, Nilesh Ahuja, and Utku Genc. Anomalib: A Deep Learning Library for Anomaly Detection. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 1706–1710. IEEE, 2022. 6

[3] Jaehyeok Bae, Jae-Han Lee, and Seyun Kim. PNI: Industrial Anomaly Detection Using Position and Neighborhood Information. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6373–6383, 2023. 3

[4] Kilian Batzner, Lars Heckler, and Rebecca König. EfficientAD: Accurate Visual Anomaly Detection at Millisecond-Level Latencies. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 128–138, 2024. 1, 2, 3, 5, 6

[5] Paul Bergmann, Sindy Löwe, Michael Fauser, David Sattlegger, and Carsten Steger. Improving Unsupervised Defect Segmentation By Applying Structural Similarity to Autoencoders. *ArXiv*, abs/1807.02011, 2018. 2

[6] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec AD–A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9592–9600, 2019. 2, 5, 1

[7] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Uninformed Students: Student-Teacher Anomaly Detection With Discriminative Latent Embeddings. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4183–4192, 2020. 2, 3

[8] Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, and Carsten Steger. Beyond Dents and Scratches: Logical Constraints in Unsupervised Anomaly Detection and Localization. *International Journal of Computer Vision*, 130 (4):947–969, 2022. 2

[9] Yue Cao, Thomas Andrew Geddes, Jean Yee Hwa Yang, and Pengyi Yang. Ensemble Deep Learning in Bioinformatics. *Nature Machine Intelligence*, 2(9):500–508, 2020. 3

[10] Yuanhong Chen, Yu Tian, Guansong Pang, and Gustavo Carneiro. Deep One-Class Classification via Interpolated Gaussian Descriptor. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 383–392, 2022. 3

[11] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. Padim: A Patch Distribution Modeling Framework for Anomaly Detection and Localization. In *International Conference on Pattern Recognition*, pages 475–489. Springer, 2021. 2, 3, 5

[12] Hanqiu Deng and Xingyu Li. Anomaly Detection via Reverse Distillation From One-Class Embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9737–9746, 2022. 2, 5

[13] André Luiz Vieira e Silva, Francisco Simões, Danny Kowerko, Tobias Schlosser, Felipe Battisti, and Veronica Teichrieb. Attention Modules Improve Image-Level Anomaly Detection for Industrial Inspection: A Differnet Case Study. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 8246–8255, 2024. 2

[14] Matic Fučka, Vitjan Zavrtanik, and Danijel Skočaj. TransFusion–A Transparency-Based Diffusion Model for Anomaly Detection. *arXiv preprint arXiv:2311.09999*, 2023. 2

[15] Mudasir A Ganaie, Minghui Hu, AK Malik, M Tanveer, and PN Suganthan. Ensemble Deep Learning: A Review. *Engineering Applications of Artificial Intelligence*, 115:105151, 2022. 3

[16] Denis Gudovskiy, Shun Ishizaka, and Kazuki Kozuka. Cflow-AD: Real-Time Unsupervised Anomaly Detection With Localization via Conditional Normalizing Flows. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 98–107, 2022. 1, 2, 3

[17] Haitian He, Sarah Erfani, Mingming Gong, and Qiuhong Ke. Learning Transferable Representations for Image Anomaly Localization Using Dense Pretraining. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1113–1122, 2024. 2

[18] Lars Heckler. and Rebecca König. Feature Selection for Unsupervised Anomaly Detection and Localization Using Synthetic Defects. In *Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 3: VISAPP*, pages 154–165. INSTICC, SciTePress, 2024. 2, 8

[19] Lars Heckler, Rebecca König, and Paul Bergmann. Exploring the Importance of Pretrained Feature Extractors for Unsupervised Anomaly Detection and Localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2916–2925, 2023. 2, 3, 6, 8

[20] Jingtao Hu, En Zhu, Siqi Wang, Xinwang Liu, Xifeng Guo, and Jianping Yin. An Efficient and Robust Unsupervised Anomaly Detection Method Using Ensemble Random Projection in Surveillance Videos. *Sensors*, 19(19):4145, 2019. 3

[21] Jeeho Hyun, Sangyun Kim, Giyoung Jeon, Seung Hwan Kim, Kyunghoon Bae, and Byung Jun Kang. ReConPatch: Contrastive Patch Representation Learning for Industrial Anomaly Detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2052–2061, 2024. 3

[22] Teng-Yok Lee, Yusuke Nagai, and Akira Minezawa. Memory-Efficient and Gpu-Oriented Visual Anomaly Detection With Incremental Dimension Reduction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2907–2915, 2023. 1, 3

[23] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-Supervised Learning for Anomaly Detection and Localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9664–9674, 2021. 2, 3

[24] Zhikang Liu, Yiming Zhou, Yuansheng Xu, and Zilei Wang. Simplenet: A Simple Network for Image Anomaly Detec-

tion and Localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20402–20411, 2023. 1, 2

[25] Peyman Nejat, Areej Alsaafin, Ghazal Alabtah, Nneka I. Comfere, Aaron Mangold, Dennis Murphree, Patricija Zot, Saba Yasir, Joaquin J. Garcia, and Hamid R. Tizhoosh. Creating An Atlas of Normal Tissue for Pruning Wsi Patching Through Anomaly Detection. *Scientific Reports*, 14(3932), 2024. 2, 3

[26] Chaewon Park, MyeongAh Cho, Minhyeok Lee, and Sangyoun Lee. Fastano: Fast Anomaly Detection via Spatio-Temporal Patch Transformation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2249–2259, 2022. 1, 3

[27] Shebuti Rayana and Leman Akoglu. Less is More: Building Selective Anomaly Ensembles. *ACM Trans. Knowl. Discov. Data*, 10(4), 2016. 3

[28] Nicolae-Cătălin Ristea, Neelu Madan, Radu Tudor Ionescu, Kamal Nasrollahi, Fahad Shahbaz Khan, Thomas B Moeslund, and Mubarak Shah. Self-Supervised Predictive Convolutional Attentive Block for Anomaly Detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13576–13586, 2022. 2

[29] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards Total Recall in Industrial Anomaly Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14318–14328, 2022. 2, 3, 4, 5

[30] Marco Rudolph, Tom Wehrbein, Bodo Rosenhahn, and Bastian Wandt. Fully Convolutional Cross-Scale-Flows for Image-Based Defect Detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1088–1097, 2022. 2

[31] Marco Rudolph, Tom Wehrbein, Bodo Rosenhahn, and Bastian Wandt. Asymmetric Student-Teacher Networks for Industrial Anomaly Detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2592–2602, 2023. 2

[32] Kuldeep Singh, Shantanu Rajora, Dinesh Kumar Vishwakarma, Gaurav Tripathi, Sandeep Kumar, and Gurjit Singh Walia. Crowd Anomaly Detection Using Aggregation of Ensembles of Fine-Tuned Convnets. *Neurocomputing*, 371:188–198, 2020. 3

[33] Kihyuk Sohn, Chun-Liang Li, Jinsung Yoon, Minho Jin, and Tomas Pfister. Learning and Evaluating Representations for Deep One-Class Classification. In *International Conference on Learning Representations*, 2021. 3

[34] Chetan L Srinidhi, Ozan Ciga, and Anne L Martel. Deep Neural Network Models for Computational Histopathology: A Survey. *Medical Image Analysis*, 67:101813, 2021. 3

[35] Chin-Chia Tsai, Tsung-Hsuan Wu, and Shang-Hong Lai. Multi-Scale Patch-Based Representation Learning for Image Anomaly Detection and Segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3992–4000, 2022. 3

[36] Guodong Wang, Shumin Han, Errui Ding, and Di Huang. Student-Teacher Feature Pyramid Matching for Anomaly Detection. In *The British Machine Vision Conference (BMVC)*, 2021. 2

[37] Guodong Wang, Yunhong Wang, Jie Qin, Dongming Zhang, Xiuguo Bao, and Di Huang. Unilaterally Aggregated Contrastive Learning With Hierarchical Augmentation for Anomaly Detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6888–6897, 2023. 2

[38] Minghui Yang, Peng Wu, and Hui Feng. Memseg: A Semi-Supervised Method for Image Surface Defect Detection Using Differences and Commonalities. *Engineering Applications of Artificial Intelligence*, 119:105835, 2023. 2

[39] Jihun Yi and Sungroh Yoon. Patch Svdd: Patch-Level Svdd for Anomaly Detection and Segmentation. In *Proceedings of the Asian conference on computer vision*, 2020. 3

[40] Jiawei Yu, Ye Zheng, Xiang Wang, Wei Li, Yushuang Wu, Rui Zhao, and Liwei Wu. Fastflow: Unsupervised Anomaly Detection and Localization via 2D Normalizing Flows. *arXiv preprint arXiv:2111.07677*, 2021. 1, 2, 5

[41] Yumna Zahid, Muhammad Atif Tahir, Nouman M Durrani, and Ahmed Bouridane. IBaggedFCNet: An Ensemble Framework for Anomaly Detection in Surveillance Videos. *IEEE Access*, 8:220620–220630, 2020. 3

[42] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. Draema Discriminatively Trained Reconstruction Embedding for Surface Anomaly Detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8330–8339, 2021. 2

[43] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. Reconstruction By Inpainting for Visual Anomaly Detection. *Pattern Recognition*, 112:107706, 2021. 2

[44] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. DSR–A Dual Subspace Re-Projection Network for Surface Anomaly Detection. In *European conference on computer vision*, pages 539–554. Springer, 2022. 2

[45] Xuan Zhang, Shiyu Li, Xi Li, Ping Huang, Jiulong Shan, and Ting Chen. Destseg: Segmentation Guided Denoising Student-Teacher for Anomaly Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3914–3923, 2023. 2

[46] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. SPot-the-Difference Self-Supervised Pre-Training for Anomaly Detection and Segmentation. In *European Conference on Computer Vision*, pages 392–408. Springer, 2022. 2, 5, 1