Exploring the Importance of Pretrained Feature Extractors for Unsupervised Anomaly Detection and Localization

Lars Heckler[1,2] Rebecca König[1] Paul Bergmann[1]

[1]MVTec Software GmbH, [2]Technical University of Munich

{lars.heckler, rebecca.koenig, paul.bergmann}@mvtec.com

Abstract

Modeling the distribution of descriptors obtained by pretrained feature extractors is a popular approach for unsupervised visual anomaly detection. While recent work primarily focuses on the development of new methods that build on such extractors, the importance of the selected feature space itself has not been sufficiently studied. We therefore conduct a systematic analysis of current anomaly detection methods with respect to different feature extractors, their intermediate layers, and pretraining protocols. We show that the investigated methods are highly sensitive to the particular choice of feature space. We further demonstrate that using an optimal feature selection strategy can significantly improve the anomaly detection performance, up to a point where selecting a single feature layer outperforms computationally expensive ensembling approaches.

1. Introduction

The detection and precise localization of anomalous structures in natural image data is an important and challenging problem in computer vision. It has applications in various domains such as medical imaging [25, 36], autonomous driving [7, 16], video surveillance [20, 27], or industrial inspection [4, 5]. Since anomalous training data is often very difficult or even impossible to acquire [3], a lot of effort is put into tackling the Anomaly Detection (AD) problem in an unsupervised way, in which a model is only given access to a training set of exclusively anomaly-free images.

Pretrained deep feature extractors have become an essential building block in many recent unsupervised AD approaches [6, 11, 26, 31, 35]. Such extractors are trained on a very large dataset of natural images to solve a certain pretext task that is not related to the AD problem. Most commonly, networks trained for image classification on the ImageNet dataset [18] are employed, which are readily available in current deep learning frameworks.

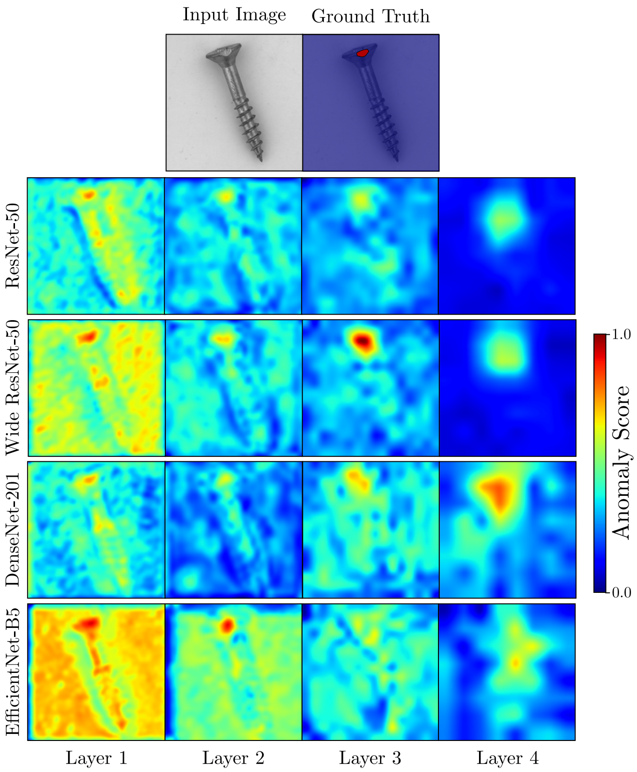

Figure 1. Dependence of the anomaly detection performance of PatchCore [32] on the underlying feature space using the example of an image of a defective screw. The predicted anomaly maps vary greatly depending on which feature extractor and intermediate feature layer is chosen.

A key reason for these generic feature extractors being so widely used is their ability to differentiate between normal and anomalous data by producing distinct features for the two classes. This characteristic has led to the development of robust unsupervised AD techniques that model the corresponding feature distribution of the anomaly-free training data.

Surprisingly, a lot of effort has been directed towards the design of new AD models that build upon such pretrained feature spaces, whereas very little work has explored the importance of the employed feature extractors. Currently, a large number of different extractors exists that could potentially be used in such AD systems. What is worse, each extractor typically contains hundreds of distinct feature layers, from which one must select a small subset – often only a single layer. To overcome this problem, it is common practice to empirically select feature layers that work well on the investigated problem. In particular, the research community has neither agreed on using one specific feature extractor, nor on a standardized feature selection strategy. This results in a wide variety of different backbones being used across research projects.

This raises the question how sensitive existing methods are to the particular choice of feature space that they operate on. To answer this, we present a unified comparison of three state-of-the-art AD approaches with respect to different pretrained feature extractors. Our findings suggest that there is a significant dependency on the used feature extractor and layer, which is qualitatively illustrated in Figure 1. Besides, one can substantially improve the AD performance by carefully selecting the appropriate feature space for a given dataset.

In particular, our key contributions are:

- We perform the first systematic analysis of the dependence of AD methods on different feature extractors, their intermediate layers, and pretraining protocols. Our results indicate that existing methods tend to be highly sensitive to these parameters and carefully choosing an appropriate feature space is essential to create accurate AD systems.

- We show that using an optimal feature selection strategy for different object categories within a dataset leads to significant performance gains. Our findings motivate a new research direction within the academic field of anomaly detection, i.e., the development of methods that determine the best performing feature layers as a function of the training data. In particular, the optimal selection of a single layer yields AD results that are on par with computationally intensive ensembling approaches.

2. Related Work

In recent years, a lot of work has been published on unsupervised anomaly detection and localization. Liu et al. [21] and Pang et al. [28] give a comprehensive overview. Regarding AD on natural images, existing methods can be broadly categorized as either based on reconstruction approaches such as convolutional autoencoders [24] or based on feature extraction from pretrained networks. In this paper, we focus on the latter.

Methods based on Pretrained Extractors. These methods attempt to model the distribution of descriptors extracted from the anomaly-free images given a pretrained backbone that is kept frozen during the entire AD process. During inference, anomalies are detected as deviations from this feature distribution under the assumption that the pretrained extractor produces different features for anomalous test images.

One line of research fits traditional machine learning models to the extracted descriptors. Cohen and Hoshen [11] model the feature distribution using k-nearest neighbors and compute anomaly scores as the distance to the nearest descriptors from the training set. The current state-of-the-art method PatchCore [32] extends this idea by an additional coreset subsampling step [37] to reduce the number of descriptors that need to be stored. In PADIM [12], the distribution of multiple feature layers is fitted with a unimodal Gaussian distribution. Reiss et al. [30] further adapt the pretrained features to the specific dataset before computing anomaly scores.

To circumvent the need for downsampling the potentially very large number of training feature vectors that is required to enable the use of shallow machine learning techniques, it has become popular to employ student–teacher models for anomaly detection [4, 6, 34, 35]. The key idea is to train a randomly initialized student network to match the descriptors extracted by a pretrained teacher network. These methods can be combined with normalizing flow modules [33] that further improve the performance. In our study, we consider the recently introduced Asymmetric Student Teacher (AsymST) [34] and FastFlow [42] methods as popular representatives from this class.

Feature Extractor Selection. All of the above methods rely on the selection of specific feature extractors, feature layers, and pretraining protocols. Interestingly, the research community has not converged to the use of a single well-performing feature extractor. For instance, PatchCore by default uses Wide ResNet-50 [43], AsymST recommends the use of EfficientNet-B5 [38], whereas FastFlow reports best results for the CaiT-48-distilled transformer architecture [39]. While some papers conduct ablation studies regarding the choice of feature extractor and feature layer [8, 13, 27, 31, 32], the investigated backbones often differ across research projects, which makes a direct comparison difficult.

Many of the aforementioned methods additionally combine features from different layers and extractors to further enhance the AD performance. For example, the best-performing PatchCore model uses an ensemble of features extracted from a DenseNet-201 [17], a ResNeXt-101 [40], and a Wide ResNet-101 [43]. Unfortunately, such ensemble methods increase the size of the search space for suitable layer combinations exponentially and come at a significant computational overhead, which may prevent their employment in real-world applications.

Feature Extractor Pretraining. To obtain descriptive general-purpose features that can be used within an anomaly detection system, a pretraining on a large dataset of natural images is usually performed. The de facto standard is to use classification networks trained on ImageNet in a supervised way. While these features were shown to transfer well to the AD problem, it has been hypothesized that high-level features from deeper layers may be biased towards the particular ImageNet classes [32], which could harm the AD performance.

An alternative pretraining strategy is to use self-supervised representation learning techniques. Popular examples include MoCo [14], SwAV [9], SeLa [2], and SimCLR [10]. These can be easily scaled to very large image databases and have proven useful for unsupervised AD [19, 22, 41, 44]. However, it is currently unclear how these self-supervised protocols compare to the supervised baseline when used in feature-based AD methods.

Against this background, it is challenging to assess the impact of the chosen feature space on the AD performance. This motivates us to conduct a unified analysis across different methods. Our study focuses on key hyperparameters such as the pretrained backbone, the intermediate feature layer, and the pretraining protocol.

3. Investigating the Importance of Feature Extractors: A Roadmap

This section describes the structure of our analysis and the parameters we examine in detail. Figure 2 shows a schematic overview of the problem setting. We are interested in the analysis of anomaly detection methods that operate on descriptors extracted from a single feature layer of a pretrained network. Given a set of network layers $\mathcal{L}$ that extract feature maps from an input image, the AD method is parameterized via the function $L^* = \text{select}(\mathcal{L})$ that selects a single element L^* from the set of available layers. In the following, we study the effect of this selection function and demonstrate its significant effect on the AD performance.

We begin our study by motivating the use of the relative receptive field of feature layers as an important characteristic to make layers within and across feature extractors comparable. Next, we introduce our experimental setup that allows a unified comparison of AD methods with respect to the underlying feature space. We further design experiments to demonstrate that estimating a suitable selection function from the training dataset is a promising avenue for future research since such selection strategies can significantly improve the AD performance. Finally, we compare the de facto standard of using a supervised protocol to pretrain the feature extractor with alternative self-supervised approaches.

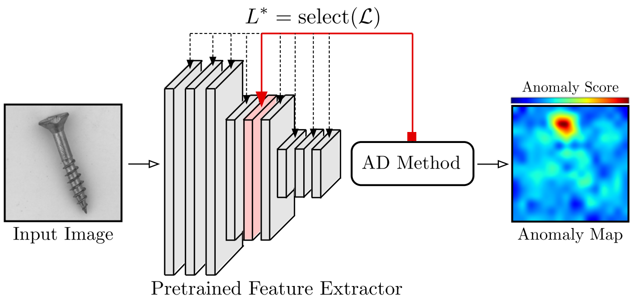
$L^* = \text{select}(\mathcal{L})$
Input Image
Pretrained Feature Extractor
AD Method
Anomaly Score
Anomaly Map

Figure 2. Investigated setup of a feature-extraction-based anomaly detection system. A single layer is selected from a certain feature extractor to be used within the AD method.

3.1. Quantifying Semantics by Receptive Field Size

Layers from varying depth within a neural network produce feature maps with very different characteristics. The spatial resolution of the feature maps often differ in addition to their capability to express semantic information. Earlier layers tend to contain more fine-grained, low-level information, whereas deeper layers predominantly capture high-level, abstract semantics. These characteristics are hypothesized to impact the AD performance [8, 32]. However, this effect has not yet been thoroughly studied since it is challenging to compare the amount of semantic information each layer captures. To gain insights into this hypothesis, we propose to order feature layers by the size of their receptive field. This allows us to compare layers with respect to the area of the input that they are sensitive to. While low-level features focus on a small area of the input, high-level semantic features capture long-range dependencies that require a considerably larger receptive field.

Following Luo et al. [23], we estimate the receptive field of a layer based on the size of the input region that effectively influences the activations of this layer. This is in contrast to computing the theoretical receptive field, which tends to overestimate the amount of pixels that actually contribute to a feature [23]. Detailed information regarding the gradient-based computation of the receptive field of feature layers is given in our supplementary material. Since we are interested in the fraction of the input image that affects a certain feature, we divide the absolute size of the receptive field in pixels by the dimensions of the input image, which yields the size of the relative receptive field (RRF).

3.2. Varying Both Feature Extractor and Layer

To study the importance of the underlying feature space for the AD performance, we vary the select function to obtain different layers L^* from various feature extractors.

We consider three state-of-the-art AD methods: PatchCore [32], FastFlow [42] and AsymST [34]. We assess the effect on both anomaly classification and anomaly localization performance to identify potential discrepancies between those measures. As feature extractors, we investigate ResNet50 [15], Wide ResNet-50 [43], DenseNet-201 [17], and EfficientNet-B5 [38], all pretrained on the ImageNet classification task. These network architectures are widely used in recent AD approaches.

For each feature extractor, we consider four distinct layers. These layers are sampled from the set of all possible layers $\mathcal{L}$ such that they are distributed evenly over the RRF. In all our experiments, exactly one layer is selected for analysis. Therefore, in total the AD performance is evaluated for 16 different layers for each method.

Object Dependency. Anomaly detection datasets often comprise multiple independent object categories. The popular MVTec Anomaly Detection Dataset (MVTec AD) [5], for instance, contains 15 distinct categories of manufactured objects. A separate anomaly detection system is trained on each of them and the resulting performance measures are averaged. Typically, AD methods employ the same feature extractor across all categories of a dataset. In our study, we are interested in how dependent the performance on individual object categories is on the selected feature space and if different feature extractors should be used for different dataset classes. To this end, we identify the best-performing layer for each dataset category and observe if a significant variance occurs.

Image Size Dependency. The size of an input image to a convolutional neural network directly correlates with the spatial resolution of the resulting feature map. Since the absolute receptive field does typically not depend on the input size, the RRF is reduced for increasing input dimensions. This, in turn, may affect the AD performance, since the amount of semantic information within a layer is also affected [8, 13]. Therefore, we study the effect of varying the input image size as well.

Optimal Feature Selection Functions. Many AD methods combine feature maps from various layers or form an ensemble of different extractors to reach high AD performance [8, 32]. While obtaining state-of-the-art results justifies this strategy, it comes with a higher computational cost that may prevent it from being employed in applications with strict runtime and memory requirements.

Since using only a single extractor and layer is usually more efficient, we raise the question if it is possible to match the performance of ensemble-based methods by carefully selecting a single layer that is most appropriate to the specific AD problem. To answer this question, we design a series of oracle selection functions that gradually assume more knowledge about the optimal feature selection strategy. We begin by computing the expected AD performance when sampling a feature layer randomly from the set of all available layers and extractors. We then assume knowledge about the feature extractor with the best mean performance and compute the expected AD performance sampling only from the layers of this extractor. Next, we additionally assume knowledge about the single layer that yields the best average performance across a dataset. This is compared to a selection strategy where the single best layer for each object category is known. Finally, we compute the performance that could be reached by using the optimal layer for each dataset object as well as the optimal image size.

This hypothetical optimal feature selection framework provides insights in the potential that lies in adaptive selection functions that are derived from the anomaly-free training data. In particular, our experiments show that selecting a single object-specific feature layer matches the performance of computationally expensive ensemble approaches.

3.3. The Importance of Supervised Pretraining

The de facto standard for feature-based AD methods is to use backbones that were pretrained on the ImageNet dataset in a fully supervised way. However, self-supervised pretraining strategies start to show impressive levels of performance in many downstream tasks, and some works already begin to integrate them into their AD systems. We are therefore interested if some pretraining protocols are more suitable for the anomaly detection downstream task than others. In particular, given a specific layer L^* from the overall set of layers $\mathcal{L}$, we assess how the pretraining protocol influences the expressiveness of the corresponding feature map and the final AD performance.

For this analysis, we reduce the overall set of possible layers to one particular feature extractor, ResNet-50 [15], for which weight initializations from a diverse set of pretraining strategies are readily available. As a baseline, we compare against the supervised ImageNet pretraining. We then test weights obtained from the following self-supervised pretraining paradigms: SimCLR [10], MoCo [14], SwAV [9] and SeLa [2]. To enable a fair comparison, each of these use the ImageNet dataset as well. In addition, we compare against a random weight initialization.

4. Experiments and Results

We conduct extensive experiments on the frequently used MVTec AD anomaly detection dataset, which comprises 15 categories of industrially manufactured objects that need to be inspected for various defects. For AsymST [34] and PatchCore [32], we build on the publicly available

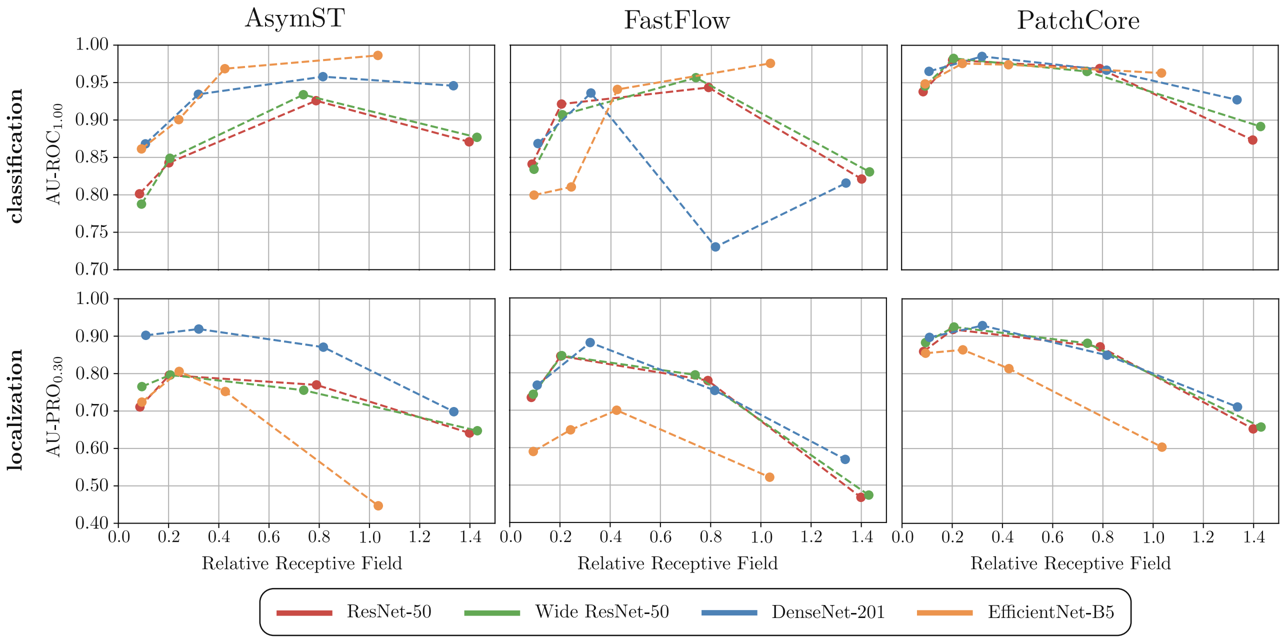

Figure 3. AD performance on MVTec AD of the evaluated methods for distinct feature spaces. Varying the feature extractor and the intermediate layer results in significant performance changes of both anomaly classification and localization.

code bases[1,2] from the original authors. For FastFlow [42], we use the implementation available in the anomalib library [1]. We extend the implementations such that various feature extractors and pretraining protocols can be tested. If not stated otherwise, we initialize all feature extractors using the official PyTorch [29] weights. We analyze four different backbone architectures: ResNet-50, Wide ResNet-50, DenseNet-201, and EfficientNet-B5. From each backbone, we extract four layers. The exact layer identifiers and further details on weight initializations for all our experiments are listed in the supplementary material.

To study the effect of different feature spaces on the anomaly detection performance, we fix the hyperparameters of the AD methods and only vary the selected layer of the feature extractor. For all evaluated methods, we set the parameters to the values recommended by the original authors except for the following changes. For PatchCore, we sample a fixed number of 1000 coreset features in each experiment instead of sampling a fraction of all available training features. Hence, the method is not sensitive to the spatial resolution of the chosen feature layer and the number of training samples. For AsymST, we reduce the number of sub-epochs to 40 for faster training. For FastFlow, we disable early stopping and train for a fixed number of 200 optimization steps. If not mentioned otherwise, all methods process images that are resized to a fixed side length of 256 × 256 pixels. We do not apply center cropping and normalize images using the official ImageNet statistics.

To compute the AD performance on MVTec AD, we use the evaluation scripts[3] released by the dataset authors [5]. This requires the anomaly images to finally match the resolution of the original dataset images, for which we use bilinear upsampling. The performance in terms of image level classification is measured by the area under the receiver operating characteristics curve (AU-ROC$_{1.00}$). The localization quality is evaluated by integrating the per-region-overlap curve to a false positive rate of 0.3 (AU-PRO$_{0.30}$).

4.1. Sensitivity on Feature Extractor and Layer

The AD performance of all evaluated methods when varying the selected feature extractor and intermediate feature layer is shown in Figure 3. The four layers of each feature extractor are ordered by their relative receptive field (RRF). Layers with larger RRFs stem from deeper hierarchy levels of the respective feature extractor. The top and bottom row show the classification and localization performance, respectively.

The particular choice of feature space has a significant impact on the AD performance for all methods. For AsymST, FastFlow, and PatchCore, the difference between the best and worst classification scores across extractors and layers is approximately 20%, 25%, and 10%, respectively. The corresponding localization scores even vary by 45%,

[1] github.com/marco-rudolph/ast
[2] github.com/amazon-science/patchcore-inspection
[3] mvtec.com/company/research/datasets/mvtec-ad

40%, and 30%.

Figure 3 further shows that there is no single best-performing feature extractor. While EfficienNet-B5 generally achieves good results for anomaly classification, its anomaly localization accuracy significantly falls behind the other extractors. DenseNet-201 in particular achieves robust localization scores across all methods. Interestingly, this also indicates that a high classification accuracy does not necessarily imply accurate anomaly localization, and vice versa. Such differences can also be observed for individual feature layers that differ in the size of their RRF. For AsymST, for example, selecting the EfficientNet-B5 layer with the largest RRF results in the highest AU-ROC$_{1.00}$, whereas the best AU-PRO$_{0.30}$ is obtained using an early DenseNet-201 layer with a significantly smaller RRF.

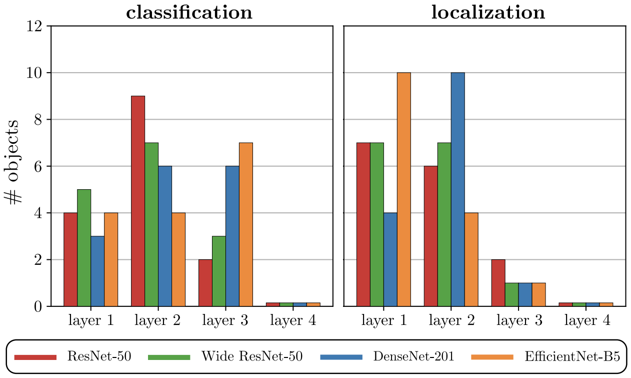

Figure 4. Number of object categories from MVTec AD for which an intermediate layer yields the best performance for PatchCore. For each feature extractor, the layers are ordered by their RRF from small to large.

Variations over Distinct Object Categories. Our experiments above report the AD performance averaged over all 15 objects of MVTec AD. To analyze the influence of the underlying feature space on the distinct categories, Figure 4 shows the number of objects for which a particular layer yields the best performance for PatchCore [32]. For both anomaly classification and localization, the best-performing layer depends on the inspected object. Again, deeper layers tend to perform better for classification, while earlier layers are more often the optimum choice for localization. We obtain similar results for AsymST [34] and FastFlow [42], which are provided in the supplementary material. Surprisingly, for PatchCore the last of the investigated layers never performs best, while for other methods this layer may lead to the top performance on average, e.g., for classification with EfficientNet-B5 with AsymST and FastFlow (cf. Figure 3). This highlights once more that extractors behave differently depending on the method in which they are used.

In general, Figure 4 indicates that the optimal extractor and layer strongly depends on the specific object category. From a practical point of view, this implies that a certain choice of extractor and layer does not necessarily generalize well to new application scenarios. An object-specific selection strategy for feature extractor and layer may have the potential to mitigate this problem.

Influence of Image Size. To analyze the influence of the input image size on the AD performance, we re-evaluate all methods with varying input dimensions. In addition to our default image size of 256×256 pixels, we test image sizes of 384×384 and 512×512 pixels. Figure 5 depicts the performance for AsymST when using WideResNet-50 and DenseNet-201 as feature extractors. Results for FastFlow and PatchCore are found in our supplementary material.

Since the RRF decreases with increased input size, the curves shift in the negative x-direction. At the same time, we notice that the performances of the layers are also affected, which confirms that the input image size is another important parameter to consider for feature-extraction-based AD. Curiously, for different image dimensions the performance of layers with similar RRF remains comparable. This indicates the tendency that the performance of a feature layer is rather defined by its RRF than by its architectural hierarchy level. For instance, for WideResNet-50, the localization performance peaks for the second layer with image size 256, but for the third layer with image sizes 384 and 512. A similar trend can also be observed for the other evaluated methods and extractors, as shown in the supplementary material.

Based on these results, we hypothesize that there is a correlation between the RRF of a layer and its respective AD performance. A possible cause for such a correlation may be the size of the anomalies within the test dataset. It seems reasonable to assume that a certain defect size is captured best on a specific hierarchy level of the convolutional feature extractor. Thus, when changing the input image size and, in doing so, the absolute defect size, the optimum RRF passes to another layer. However, we leave more fine-grained experiments to test this hypothesis for future work.

Oracle Feature Selection Functions. The results from the above experiments highlight the potential benefits that a problem-specific selection of a suitable feature extractor and intermediate layer could yield. To theoretically investigate how such an optimal selection strategy would affect the AD performance, we derive a series of oracle feature selection functions. These functions explore different parameter subsets and select the best possible combination. We use independent oracles for classification and localization and keep the image size fixed to 256×256 pixels unless stated otherwise.

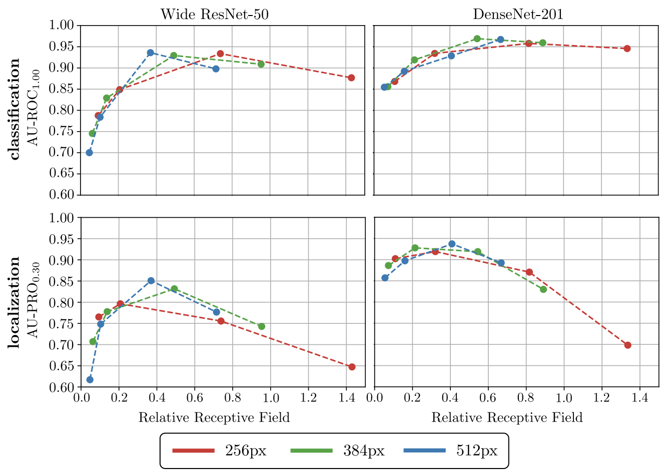

Figure 5. Varying the input image size for AsymST when using Wide ResNet-50 and DenseNet-201 as feature extractors. Increasing the input dimension reduces the RRF and also affects the performance of the individual feature layers.

Figure 6 shows the anomaly classification and localization performance for different oracle levels. As a baseline, the average performance over all layers and feature extractors is reported for each method. It represents the expected value of a random layer selection (*random*). Next, the oracle is allowed to select the most suitable feature extractor, i.e., the extractor with the highest average performance over its intermediate layers (*best feature extractor*). Already this first optimum on the extractor level leads to a considerable increase in AD performance across the examined methods. Then, the oracle selects the best of the 16 layers across all available extractors (*best layer*). As expected, this leads to further performance improvements. Additionally enabling the oracle to pick the optimal extractor and layer per object category (*best layer per object)* results in near state-of-the-art AD performance across methods. Classification AU-ROC$_{1.00}$ (localization AU-PRO$_{0.30}$) becomes 99.3% (93.8%) for AsymST, 98.3% (91.0%) for FastFlow, and 99.4% (95.3%) for PatchCore. Finally, instead of using a fixed input image size, we allow the oracle to choose the optimal input dimension from all three investigated image sizes before selecting the best layer per object (*best img size & layer per object*). This enables AsymST and PatchCore to nearly perfectly solve the image-level classification on MVTec AD with an AU-ROC$_{1.00}$of 99.7% and 99.9%, respectively.

Quantitative values for the last two oracles are also provided in Table 1. We additionally report the AD performance of the evaluated methods in their original training configuration, which includes using model ensembles with distinct feature extractors (FastFlow, PatchCore) as well as concatenating features from multiple layers (PatchCore). Although such oracles are just a theoretical construct, the ideal layer selection strategies push the AD performance towards the limits for all investigated methods. Often, the original setting is even outperformed, which demonstrates that utilizing just a single layer of a feature extractor can match the accuracy of computationally more expensive ensembling approaches. Therefore, we believe trying to automatically identify such an optimal selection strategy based on the anomaly-free training dataset is a very promising but yet unexplored research direction.

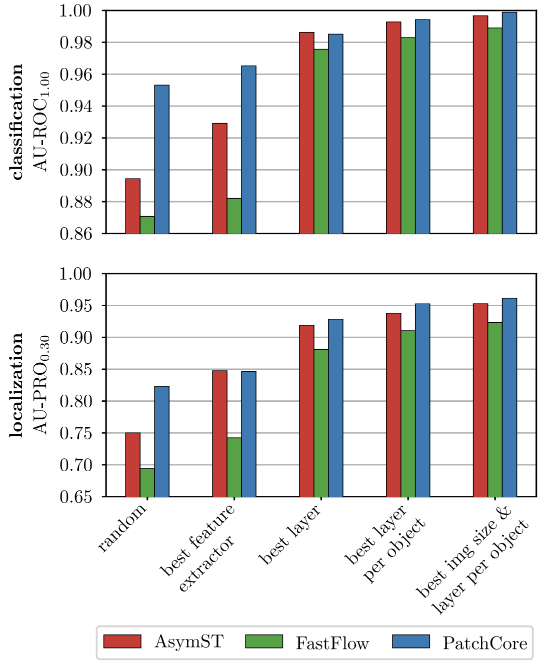

Figure 6. AD performance for different oracle feature selection functions. Successively including knowledge on best-performing parameters for feature extraction leads to state-of-the-art performances for all evaluated methods.

Table 1. Classification AU-ROC$_{1.00}$ (localization AU-PRO$_{0.30}$) for AsymST, FastFlow, and PatchCore using the two best-performing oracle feature selection functions. Additionally, the performance of the evaluated methods in their original training configuration is reported, which includes ensembling strategies if applicable.

	AsymST	FastFlow	PatchCore
best layer per object	99.3% (93.8%)	98.3% (91.0%)	99.4% (95.3%)
best img size & layer per object	**99.7%** **(95.3%)**	**98.9%** (92.3%)	**99.9%** **(96.1%)**
reproduced results	98.9% (81.2%)	96.9% **(92.5%)**	99.3% (95.5%)

4.2. Influence of Different Pretraining Strategies

Figure 7 presents the influence of different pretraining strategies when using a ResNet-50 feature extractor for FastFlow. The results for the other methods are provided in the supplementary material. Since the estimation of the RRF is gradient-based, it depends on the model weights and, thus, was recomputed for the different weight initializations. Indeed, changes of the RRF can be observed across all layers.

In general, significant performance differences occur for a single layer when using different pretraining strategies. As expected, initializing the feature extractor with random weights does not result in competitive AD performance. Supervised ImageNet pretraining significantly improves the performance over this random baseline. However, using the examined self-supervised pretraining techniques leads to comparable results, except for SimCLR.

Our experiments indicate that representations from self-supervised learning can transfer well to the anomaly detection problem and that supervised techniques are not necessarily required. However, no single pretraining paradigm consistently outperforms all others. Since there is still room for improvement, even better weight initializations for feature-extraction-based unsupervised AD may exist. Thus, one possible avenue for future research is the construction of feature extractors that are more tailored towards the anomaly detection problem itself, e.g., by pretraining on a large AD-specific dataset.

5. Conclusion

This paper investigated the importance of pretrained feature extractors in unsupervised visual anomaly detection systems. While recently a lot of new AD methods that build on pretrained feature spaces have been developed, little effort has been directed towards understanding the impact that the particular choice of feature space can have on the AD performance. To date, the community lacks systematic approaches to select a suitable feature extractor and intermediate layer and relies on empirical selections that work well for an investigated problem.

We conducted a systematic analysis of three state-of-the-art AD methods and tested their performance on 16 individual feature layers, originating from four different feature extractors. Experiments on the MVTec AD dataset reveal that all examined methods tend to be highly sensitive to the particular choice of feature space and no single best-performing feature extractor exists. We further show that the optimal feature layer may also vary with respect to the inspected object category, the input image size, and the used pretraining protocol.

Finally, we show that using an optimal feature selection strategy with respect to the distinct objects of the MVTec AD dataset can significantly improve the performance. By using only a single feature layer, we reach results up to 99.9% AU-ROC$_{1.00}$ for anomaly classification and 96.1% AU-PRO$_{0.30}$ for anomaly localization. This matches the performance of the current state of the art, which relies on computationally expensive feature ensembling techniques. This result motivates a new research direction, i.e., the development of methods that select appropriate feature layers as a function of the anomaly-free training data.

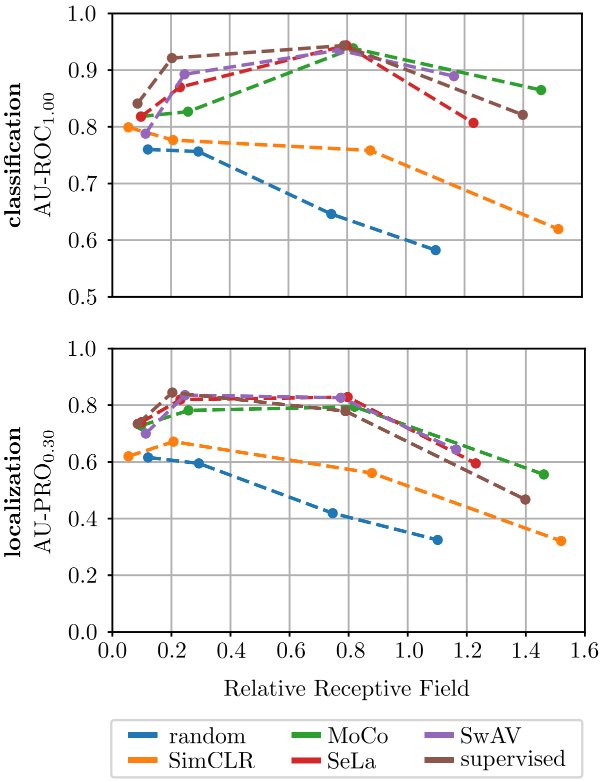

Figure 7. AD performance of FastFlow with ResNet-50 as feature extractor using different pretraining strategies. Weight initializations obtained from self-supervised paradigms are a competitive alternative to supervised ImageNet pretraining.

In future work, our studies may be continued in a variety of ways. First, the presented experiments can be repeated on additional AD datasets, e.g., VisA [44] or MVTec LOCO [4]. Second, testing the effect of even more intermediate layers within feature extractors would be of interest to see if even better AD performances can be achieved by enabling more fine-grained layer selections. Third, our work may be extended to ensembling techniques that require feature selection functions that choose multiple extractor layers. In particular, studying the effect that combining different feature layers in ensembling approaches has on the AD performance is a promising avenue for future research. As in the single-layer setting, it would be interesting to develop object-specific feature selection strategies.

References

[1] 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, 2022. 5

[2] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *8th International Conference on Learning Representations, ICLR 2020*, 2020. 3, 4

[3] Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, and Carsten Steger. The MVTec Anomaly Detection Dataset: A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. *International Journal of Computer Vision*, 129(4):1038–1059, 2021. 1

[4] 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. 1, 2, 8

[5] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. MVTec AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9584–9592, 2019. 1, 4, 5

[6] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Uninformed Students: Student-Teacher Anomaly Detection With Discriminative Latent Embeddings. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4182–4191, 2020. 1, 2

[7] Hermann Blum, Paul-Edouard Sarlin, Juan Nieto, Roland Siegwart, and Cesar Cadena. Fishyscapes: A Benchmark for Safe Semantic Segmentation in Autonomous Driving. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 2403–2412, 2019. 1

[8] Yunkang Cao, Xiaohao Xu, Zhaoge Liu, and Weiming Shen. Collaborative discrepancy optimization for reliable image anomaly localization. *IEEE Transactions on Industrial Informatics*, pages 1–10, 2023. 2, 3, 4

[9] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems*, 2020. 3, 4

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 13–18 Jul 2020. 3, 4

[11] Niv Cohen and Yedid Hoshen. Sub-image anomaly detection with deep pyramid correspondences. *arXiv preprint arXiv:2005.02357v1*, 2020. 1, 2

[12] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. Padim: A patch distribution modeling framework for anomaly detection and localization. In *Pattern Recognition. ICPR International Workshops and Challenges 2021, Proceedings, Part IV*, volume 12664 of *Lecture Notes in Computer Science*, pages 475–489. Springer, 2020. 2

[13] Denis Gudovskiy, Shun Ishizaka, and Kazuki Kozuka. Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1819–1828, 2022. 2, 4

[14] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2020. 3, 4

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 4

[16] Dan Hendrycks, Steven Basart, Mantas Mazeika, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. A Benchmark for Anomaly Segmentation. *arXiv preprint arXiv:1911.11132v1*, 2019. 1

[17] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2, 4

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification With Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012. 1

[19] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9659–9669, 2021. 3

[20] Wei-Xin Li, Vijay Mahadevan, and Nuno Vasconcelos. Anomaly Detection and Localization in Crowded Scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(1):18–32, 2013. 1

[21] Jiaqi Liu, Guoyang Xie, Jingbao Wang, Shangnian Li, Chengjie Wang, Feng Zheng, and Yaochu Jin. Deep industrial image anomaly detection: A survey. *arXiv preprint arXiv:2301.11514*, 2023. 2

[22] Jun Long, Yuxi Yang, Liujie Hua, and Yiqi Ou. Self-supervised augmented patches segmentation for anomaly detection. In *Computer Vision – ACCV 2022*, pages 93–107, Cham, 2023. Springer Nature Switzerland. 3

[23] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. 3

[24] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction. In *Artificial Neural Networks and Machine Learning – ICANN 2011*, pages 52–59. Springer, 2011. 2

[25] Bjoern H. Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, et al.

The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). *IEEE Transactions on Medical Imaging*, 34(10):1993–2024, 2015. 1

[26] Paolo Napoletano, Flavio Piccoli, and Raimondo Schettini. Anomaly Detection in Nanofibrous Materials by CNN-Based Self-Similarity. *Sensors*, 18(1):209, 2018. 1

[27] Tiago S Nazare, Rodrigo F de Mello, and Moacir A Ponti. Are pre-trained cnns good feature extractors for anomaly detection in surveillance videos? *arXiv preprint arXiv:1811.08495*, 2018. 1, 2

[28] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton van den Hengel. Deep Learning for Anomaly Detection: A Review. *arXiv preprint arXiv:2007.02500*, 2020. 2

[29] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, volume 32, 2019. 5

[30] Tal Reiss, Niv Cohen, Liron Bergman, and Yedid Hoshen. Panda: Adapting pretrained features for anomaly detection and segmentation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2805–2813, 2021. 2

[31] Oliver Rippel, Patrick Mertens, and Dorit Merhof. Modeling the distribution of normal data in pre-trained deep features for anomaly detection. In *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021. 1, 2

[32] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter V. Gehler. Towards total recall in industrial anomaly detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022*, pages 14298–14308. IEEE, 2022. 1, 2, 3, 4, 6

[33] Marco Rudolph, Tom Wehrbein, Bodo Rosenhahn, and Bastian Wandt. Fully convolutional cross-scale-flows for image-based defect detection. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2022*, pages 1829–1838. IEEE, 2022. 2

[34] Marco Rudolph, Tom Wehrbein, Bodo Rosenhahn, and Bastian Wandt. Asymmetric student-teacher networks for industrial anomaly detection. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2023*, pages 2591–2601. IEEE, 2023. 2, 4, 6

[35] Mohammadreza Salehi, Niousha Sadjadi, Soroosh Baselizadeh, Mohammad H. Rohban, and Hamid R. Rabiee. Multiresolution knowledge distillation for anomaly detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14897–14907, 2021. 1, 2

[36] Thomas Schlegl, Philipp Seeböck, Sebastian Waldstein, Georg Langs, and Ursula Schmidt-Erfurth. f-AnoGAN: Fast Unsupervised Anomaly Detection with Generative Adversarial Networks. *Medical Image Analysis*, 54, 2019. 1

[37] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *6th International Conference on Learning Representations, ICLR 2018*, 2018. 2

[38] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 2, 4

[39] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 32–42, 2021. 2

[40] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5987–5995, 2017. 2

[41] Jihun Yi and Sungroh Yoon. Patch svdd: Patch-level svdd for anomaly detection and segmentation. In *Computer Vision – ACCV 2020*, pages 375–390, Cham, 2021. Springer International Publishing. 3

[42] 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:2111.07677v2*, 2021. 2, 4, 5, 6

[43] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference 2016*. British Machine Vision Association, 2016. 2, 4

[44] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In *Computer Vision – ECCV 2022*, pages 392–408, Cham, 2022. Springer Nature Switzerland. 3, 8