Enhanced Novelty Detection in Images Using Auto-Encoders and Teacher-Student Knowledge Transfer

Parsa Mohammadrezaei
Department of Computer Science
Concordia University
parsa.mohammadrezaei@mail.concordia.ca

Farid Farahmand
Department of Computer Science
Concordia University
farid.farahmand@mail.concordia.ca

Sam Collin
Department of Computer Science
Concordia University
sam.collin@mail.concordia.ca

Abstract

Anomaly detection is a critical task in domains such as quality control and medical imaging, where deviations from normal patterns must be identified without relying on extensive abnormal data.[1] Recent advancements, like the student-teacher framework, have shown promise in this area by leveraging pre-trained models for efficient and accurate pixel-level anomaly detection. Despite their success, there is potential to further enhance the framework's ability to learn and reconstruct anomaly-free distributions.

This work is an attempt to outperform the student-teacher feature pyramid matching for anomaly detection paper [2], we address it's limitation by integrating auto-encoders into the student-teacher framework. This integration is done to better capture anomaly features to enable effective anomaly scoring. Our proposed approach aims to achieve results comparable to or better than the baseline framework on the MVTec anomaly detection dataset.

1 Introduction

Anomaly detection is the process of identifying data points or patterns that deviate significantly from expected behavior, often signaling critical or rare events across various applications.[1] This task is particularly challenging due to the limited availability of labeled abnormal data, necessitating the use of one-class learning techniques. While existing methods, such as the student-teacher framework [2], have shown promise in pixel-level anomaly detection by leveraging pre-trained models and hierarchical knowledge transfer, there remains a need for further refinement, particularly in capturing and reconstructing the fine-grained features of anomaly-free distributions.

Accurate anomaly detection plays a pivotal role in fields that demand high precision and reliability. The ability to detect subtle deviations without extensive abnormal training data is both practically valuable and scientifically intriguing. While the student-teacher framework has achieved notable success, integrating additional mechanisms for feature reconstruction and refinement could significantly enhance detection accuracy and robustness, addressing existing limitations and broadening its applicability.

In this work, we extend the student-teacher framework by integrating auto-encoders into the student network. This enhancement should enable the student to better reconstruct anomaly-free image distributions, thereby strengthening its ability to learn more robust feature representations and detect anomalies with an accuracy comparable to the baseline.

The paper is organized as follows:

- **Section 2** reviews related work on anomaly detection and the student-teacher framework.
- **Section 3** presents the details of the proposed hybrid architecture and the integration of the auto-encoders.
- **Section 4** presents a selection of the most noteworthy results from our computations, highlighting the best outcomes obtained.
- **Section 5** concludes the paper and provides directions for future work.

2 Related Work

The related work can be broadly divided into two main areas: image-level anomaly detection and pixel-level anomaly detection. We will also discuss the specific approach emphasized in the STFPM.

Image Level Anomaly Detection

Image-level anomaly detection encompasses three distinct models:

Reconstruction-based: The initial category of methods focuses on reconstructing training images to model the distribution of normal data. During inference, anomalous images typically exhibit significant reconstruction errors, as they originate from a different distribution. A major limitation of these approaches lies in the strong generalization capability of deep models, such as variational autoencoders, which may allow anomalous images to be accurately reconstructed.[3]

Distribution-based: Distribution-based approaches aim to model the probabilistic distribution of normal images, classifying images with low probability densities as anomalous. Recent techniques, such as anomaly detection GAN (ADGAN), follow this approach. However, these methods tend to suffer from high sample complexity and require large amounts of training data.[4]

Classification-based: Over the past decade, classification-based approaches have dominated anomaly detection. A common approach is to use deep features, either extracted from deep generative models or transferred from pre-trained networks, as inputs.[5]

Pixel level anomaly detection

Pixel-level techniques are specifically developed for anomaly localization. They focus on accurately identifying and segmenting anomalous regions within images, a task that is more complex than binary classification. The powerful capabilities of deep neural networks have led to numerous studies investigating how to leverage networks pre-trained for image classification to enhance anomaly detection. For instance, Napoletano et al. [6] utilize a pre-trained ResNet-18 to map cropped patches of training images into a feature space, apply PCA for dimensionality reduction, and use K-means clustering to model the feature distribution. However, this approach necessitates processing a large number of overlapping patches to generate spatial anomaly maps during inference, leading to coarse-grained maps and potentially hindering performance.

ST-FPM paper as a starting point

As mentioned above, our work is based on the work carried out by Wang et al. [2] In this paper, they employ a structure named student-teacher framework to adress pixel-level anomaly detection. The student-teacher method is based on the use of a complex, well-trained teacher model. This teacher serves as a reference for the student, a model whose main objective is to reproduce the teacher's predictions. They use the same architecture (e.g. three ResNet-18 blocks) for both models, which

minimizes information loss. Because of the pyramidal shape of the architecture, multi-scale feature matching is embedded. It allows combination of low and high level information to detect different types of anomalies.

First, in the training process the goal is to obtain the best student possible on anomaly-free images. This way, the student will learn for this specific task. Images at the input are $\mathbf{I}_k \in \mathbb{R}^{w \times h \times c}$ and after passing through the ℓth layer, the feature maps for the teacher is $F_t^l(\mathbf{I}_k) \in \mathbb{R}^{w_l \times h_l \times d_l}$ where d is the number of channels of the feature map. The same exists for the student. Considering feature vectors at position (i, j) along the d axis : $F_t^l(\mathbf{I}_k)_{ij} \in \mathbb{R}^{d_l}$ and $F_s^l(\mathbf{I}_k)_{ij} \in \mathbb{R}^{d_l}$. The loss is defined at position (i, j) as ℓ_2 distance between the ℓ_2 normalized feature vectors.

$$\ell^l(\mathbf{I}_k)_{ij} = \frac{1}{2}\|\hat{F}_t^l(\mathbf{I}_k)_{ij} - \hat{F}_s^l(\mathbf{I}_k)_{ij}\|_{\ell_2}^2 \tag{1}$$

Normalization is performed as :

$$\hat{F}_t^l(\mathbf{I}_k)_{ij} = \frac{F_t^l(\mathbf{I}_k)_{ij}}{\|F_t^l(\mathbf{I}_k)_{ij}\|_{\ell_2}}, \quad \hat{F}_s^l(\mathbf{I}_k)_{ij} = \frac{F_s^l(\mathbf{I}_k)_{ij}}{\|F_s^l(\mathbf{I}_k)_{ij}\|_{\ell_2}}.$$

Moving forward, the image is the average of each position and then a weighted average at different pyramid scales. Second, in the test phase, the objective is to obtain an anomaly map $\mathbf{\Omega}$ with the same size that the input test image $\mathbf{J} \in \mathbb{R}^{w \times h \times c}$. The score $\mathbf{\Omega}_{ij} \in [0, 1]$ indicates how much the pixel differs. The anomaly map $\mathbf{\Omega}^l(\mathbf{J})$ for a specific layer will be the loss function (Eq 1) at each position (i, j). The anomaly map for each layer is upsampled to size $w \times h$ and then the final multi-scale anomaly-map is the element wise product of the three equal sized anomaly maps :

$$\mathbf{\Omega}(\mathbf{J}) = \prod_{l=1}^{3} Upsample\mathbf{\Omega}^l(\mathbf{J}) \tag{2}$$

If any pixel in the image is anomalous, than it is detected as anomaly. The maximum anomaly value for a pixel in the map is considered as the anomaly score for the test image. If we disregard the auto-encoders and assume that R_s the reconstruction equals F_s the student's feature map, this architecture is represented in Figure 1.

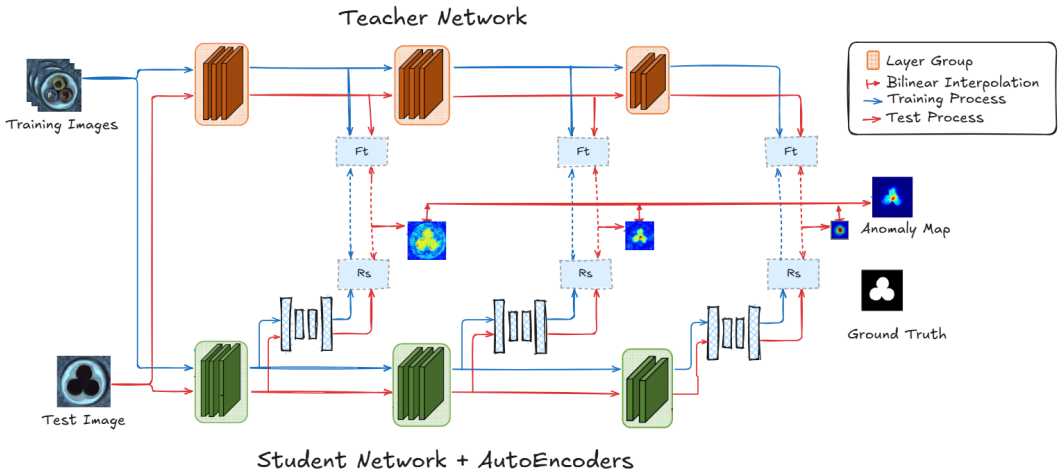

Figure 1: Schematic overview of our extended method. It adds three auto-encoders to the architecture proposed by Wang et al. [2] to reconstruct the student's feature maps at each level of the architecture. Training and testing process is represented as long as the multi-scale anomaly map computation path. Anomalies are detected in a single forward pass.

3 Methodology

Main idea

The core of our attempt to improve the anomaly detection model from the paper is to pass all the feature maps $F_s^l \in \mathbb{R}^{w_l \times h_l \times d_l}$ output by the student model through an auto-encoder at each layer. These reconstructed feature maps $R_s^l \in \mathbb{R}^{w_l \times h_l \times d_l}$ are then compared to the teacher's feature map as they were before. To do so, we are adding three different auto-encoders to the former architecture. They all share the same general architecture, but because the number of channels is not the same for all output feature maps, they are adapted to the current layer. Spatial resolution also differs between layers but is implicitly handled by the models as it remains consistent. Final architecture is showed in Figure 1. By doing this, we expect that the anomalies will be more difficult for the auto-encoder to reconstruct because they were not seen during training, leading to a larger difference from the teacher in these areas.

Choosing the right Auto-Encoder

If we want our implementation to have a chance to outperform the architecture from the ST-FPM paper. We need to choose an auto-encoder that will complete the two following objectives as well as possible :

- Reconstruct perfectly all the anomaly-free images to match the teacher's feature maps.
- Amplify the reconstruction error once it encounters unseen data (e.g. anomalies)

We tested different type of auto-encoders from basic ones to more advanced architecture, our choice finally falls on a UNET architecture [7]. The one used in our implementation differs from the original one by being lighter as we are not performing image segmentation and working already on feature maps. It is composed of a contracting path (encoder) and an expanding path (decoder) linked together by the bottleneck. The encoder is a succession of three convolutional blocks, each made of two 3x3 convolution + ReLU layer followed by a downsampling operation using 2x2 maxpooling. The encoder is connected to the decoder by the bottleneck which is a central convolutional block that maintains the same numbers of channels. Now, the expanding path in our implementation is where it gets interesting. In common UNET architecture, the decoder is mirroring the encoder. In our case, due to an error in the definition of the architecture, we realized that we could get better results simply by using an upsampling function that performs a 2x2 transposed convolution and concatenates the corresponding features of the contraction path. No additional convolution is applied after concatenation. This particular architecture is represented in Figure 2.

We suspect that this may be due to better generalization because it is a simpler model; if the characteristics are simple enough, additional convolutions may not be necessary. This also saves some computation time. Finally, as model performance is the ultimate judge of efficiency, we decided to retain this architecture rather than a more standard one.

Pre-training the Auto-Encoder

There is plenty of possibilities to implement the auto-encoder in the student teacher architecture. We tried many options, from putting it in raw to freezing it completely after a lot of pre-training. The most promising was a combination of the two in a fine-tuning approach. The auto-encoder is first trained to reconstruct the features from the teacher model. Let $F_t^l \in \mathbb{R}^{w_l \times h_l \times d_l}$ denote the feature maps obtained from the student at l-th layer and $R_s^l \in \mathbb{R}^{w_l \times h_l \times d_l}$ the reconstruction output by the auto-encoder $\mathcal{A}^l$ at layer l. The objective of the pretraining is then to minimize the sum of the MSE loss function on each layer:

$$\mathcal{L}_{pre} = \sum_l \|\mathcal{A}^l(F_t^l) - F_t^l\|_2^2$$

to optimize this loss function, we use the Adam optimizer with a learning rate of 0.001. This process ensures that the auto-encoders, once in the final architecture are well-initialized and just need some slight adjustments to complete the wanted task. Because it is only trained on anomaly-free images, they will get a strong basis on the normal pattern and will just have to adapt to the student model.

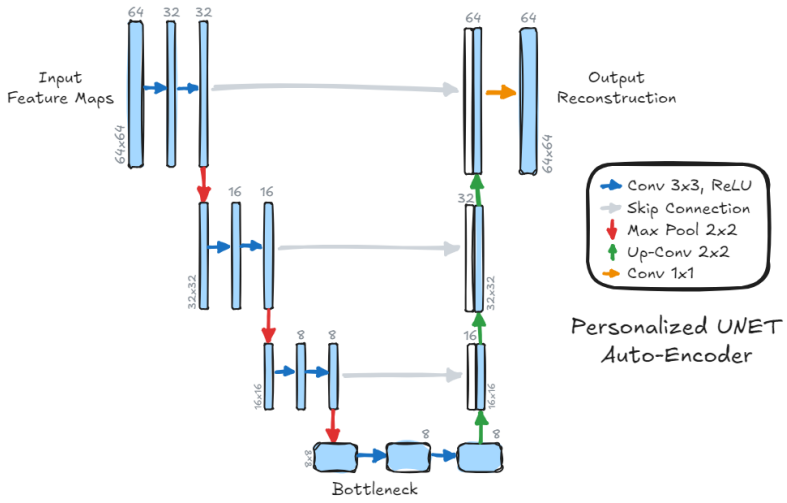

Figure 2: Schematic overview of our personalized U-net architecture. This model represents the auto-encoder for the first layer of our framework, but the idea is the same for the others. It processes the feature maps progressively, reducing the resolution by a contraction path to a bottleneck, and then reconstructing the feature maps to the original resolution and channel depth. Skip connections link the encoder and decoder and no convolution is applied after the concatenation.

Training process

The training process is based on the one from the ST-FPM paper [2], however, now that we have added three auto-encoders to the model we need to modify the loss function to take account of these changes. The total loss for an input image I_k is defined as:

$$\mathcal{L}_{tot} = \mathcal{L}_{rec} + \mathcal{L}_{sim} = \alpha \sum_l \|R_s^l(I_k) - F_s^l(I_k)\|_2^2 + \sum_l \|\phi(R_s^l(I_k)) - \phi(F_t^l(I_k))\|_2^2$$

where:

- $\mathcal{L}_{rec}$ is the reconstruction loss that helps the auto-encoder to learn specific feature maps coming from the student to reconstruct them as best as possible.

- $\mathcal{L}_{sim}$ is the similiraty loss measuring how well the student model mimics the teacher.

- $\phi(.)$ is the normalization function putting features on a comparable scale.

- α is the weighting parameter. It is fixed for now as dynamic weighting wasn't performing well. If well chosen, it allows training to be focused on reconstruction or similarity loss, depending on the objective. It is also useful to rescale the reconstruction loss. Much higher than the similarity.

Optimization is performed using SGD. In order to prevent auto-encoders that do not start from scratch from losing all their knowledge (catastrophic forgetting), we have differentiated the learning rates from those of the student model. In use, the student network uses a higher learning rate to quickly adapt to the teacher, while the auto-encoders use a lower one to ensure stability:

$$\theta_s \leftarrow \theta_s - \eta_s \nabla_{\theta s} \mathcal{L}_{tot}$$

$$\theta_{\mathcal{A}} \leftarrow \theta_{\mathcal{A}} - \eta_{\mathcal{A}} \nabla_{\theta_{\mathcal{A}}} \mathcal{L}_{tot}$$

with $\eta_{\mathcal{A}} < \eta_s$.

We also use gradient clipping to avoid gradient explosion, it helps us to limit the norm of the gradients during backpropagation and make the updates more reasonable. Finally, we saw that freezing the auto-encoders for the first five epochs was a good practice so we stuck to it afterwards. This gives the student model time to take an initial direction before influencing the auto-encoders.

Test process

During the test process, an input **x** will pass through both the frozen teacher and the student network. This allows to extract the feature maps at each layers. After that, the student's feature map F_s^l at layer l will pass through the respective auto-encoder $\mathcal{A}^l$. The anomaly map is then the loss computed for each pixel i,j. As each layer has a different resolution, we upsample each anomaly map with the bilinear interpolation to finally obtain our final multi-scale anomaly map (i.e. our prediction). This is then compared with the ground truth using the AUC-ROC metric. The advantage of the AUC-ROC score is that it allows us to compare the continuous values of our anomaly map with the binary ground truth, and in effect implicitly scans all the thresholds, giving us a very relevant overview of the performance of our model. With the exception of the auto-encoder pass, the process is the same as for the ST-FPM paper.

Implementation

For our work, we focused on the MVTEC Anomaly Detection (MVTec AD) dataset well known for anomaly localization. It consists of more than 5000 images of industrial products, divided into those with defects and those without on over 15 categories. We've focused on two categories in particular in this project: bottles and leather, with the aim of going beyond the paper. We kept the same three blocks of the ResNet-18 (i.e., conv2_x, conv3_x, conv4_x) for the pyramid extractors and the student is randomly initialized. The UNET auto-encoders have been pre-trained for 100 epochs using Adam optimizer and a learning rate of 0.001. The student network and the fine tuning of the auto-encoders have been trained for 100 epochs using the Stochastic Gradient Optimizer (SGD) with a learning rate of 0.4 and 0.04 respectively and momentum of 0.9. The weighting parameter α is set to 1e-7. The auto-encoders are freezed for the five first epochs.

4 Results

Our experimental evaluation demonstrates the efficacy of integrating a UNET-based autoencoder into the student-teacher framework for anomaly detection. We present results on the MVTec Anomaly Detection (MVTec AD) dataset, focusing on the "bottle" and "leather" categories. Table 1 & 2 summarizes the performance metrics, including Pixel-Level AUC, and the final training loss.

The results from [2] are comparable to our work. Our architectures have resulted in slightly less AUC-ROC on most model types.

The calculated anomaly maps show noticeable visual differences, with clearer and more distinct patterns compared to previous results. Our updated approach has led to visually superior outcomes, particularly in highlighting anomalies. These differences are especially evident in both the bottle (See Fig. 4) and leather (See Fig. 3 & 5) categories, where the anomalies are now more well-defined and distinct.

This shows that presence of autoencoders seems to play a role in this improvement. By capturing and reconstructing complex features more effectively, the they contribute to a better representation of normal patterns. (*It should be noted, however, that the visual representations are those obtained by the code implemented by the community. Paper does not provide code.*)

Table 1: AUC-ROC performance metrics for our model 1 using the custom UNET architecture 2 tested with different hyperparameters on bottles and leather images. (Reduced Version, see Table 2 for more details)

Model	Pixel Level AUC	Original Pixel AUC	Image Level AUC
UNET on leather	0.9914	**0.993**	0.9786
UNET on leather (001)	0.9903	**0.993**	0.9759
UNET on leather (002)	0.9292	**0.993**	0.6262
UNET on leather (003)	0.9925	**0.993**	0.9976
UNET on leather (004)	**0.9934**	0.993	0.9973
UNET on leather (005)	0.9782	**0.993**	0.9823
UNET on bottle (001)	**0.9837**	**0.998**	1.000

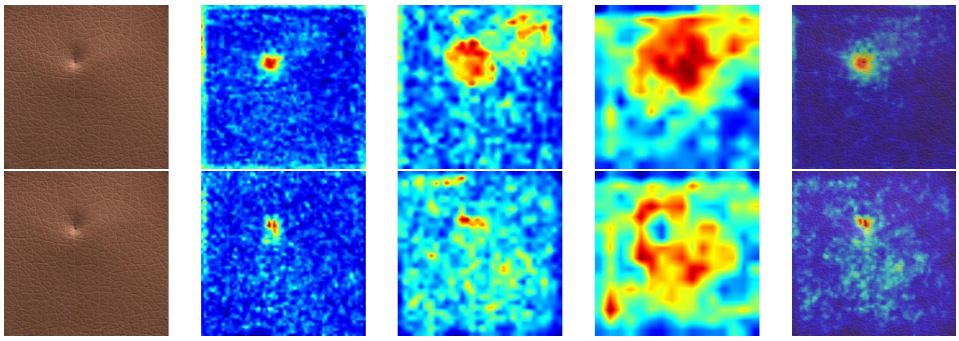

Figure 3: Visual results of our method compared to the classical ST-FPM architecture [2] on a defective image of a leather shred. The top row is our model. Columns from left to right correspond to input image, anomaly maps of the three blocks in reverse order (16x16,32x32,64x64) and the resulting anomaly map superposed on the image.

5 Discussion and Conclusion

This work represents a really interesting way to learn more about anomaly detection. Incorporating personalized U-net auto-encoders into a student-teacher feature pyramid matching architecture to enhance it's ability to detect anomalies at different levels of detail. The results confirm the approach, with performance metrics equal to, and in one case slightly better than, the baseline from the paper model. These are demonstrations that there is potential behind this approach of adding a specialized auto-encoder layer before comparison. However, we need to be cautious and realize that this is not being done for free, but at the cost of a trade-off that needs to be taken seriously.

The major compromise here comes from the added complexity. Although the results look promising, they are not enough to definitively justify the increase in complexity in terms of calculation and architecture. This highlights a very important lesson in deep learning: a more complex, deeper, more highly trained model will not always be better. Of course, this should not detract from the value of our work, but we would like to point out that we are aware of the potential for unnecessary complexity that would come with this area of improvement. With this project, we immersed ourselves in a more research-oriented approach, learning to juggle implementations, tests, reading, and theoretical understanding as we gradually wrote this paper.

For the future, there are a number of areas that we haven't had time to explore further. Reconstruction loss, for example, could be normalized to provide better guidance for auto-encoders. We could also deal with the border effect during convolutions by using cropping methods between layers. This would make it possible to limit this effect and perhaps improve accuracy even further. The weighting parameter alpha is fixed for the moment, because it's complicated to set properly, but it could be interesting to make it dynamic in order to better manage our training phase. Finally, we could continue to try out different combinations indefinitely in order to optimize our hyper-parameters as much as possible.

References

[1] 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] Guodong Wang, Shumin Han, Errui Ding, and Di Huang. Student-teacher feature pyramid matching for anomaly detection, 2021.

[3] Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. Technical report, SNU Data Mining Center, 2015.

[4] Lucas Deecke, Robert Vandermeulen, Lukas Ruff, Stephan Mandt, and Marius Kloft. Image anomaly detection with generative adversarial networks. In *ECML-PKDD*, pages 3–17, 2018.

[5] Philippe Burlina, Neil Joshi, and I-Jeng Wang. Where's wally now? deep generative and discriminative embeddings for novelty detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[6] Paolo Napoletano, Flavio Piccoli, and Raimondo Schettini. Anomaly detection in nanofibrous materials by cnn-based self-similarity. *Sensors*, 18(2):209, 2018.

[7] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

Appendices

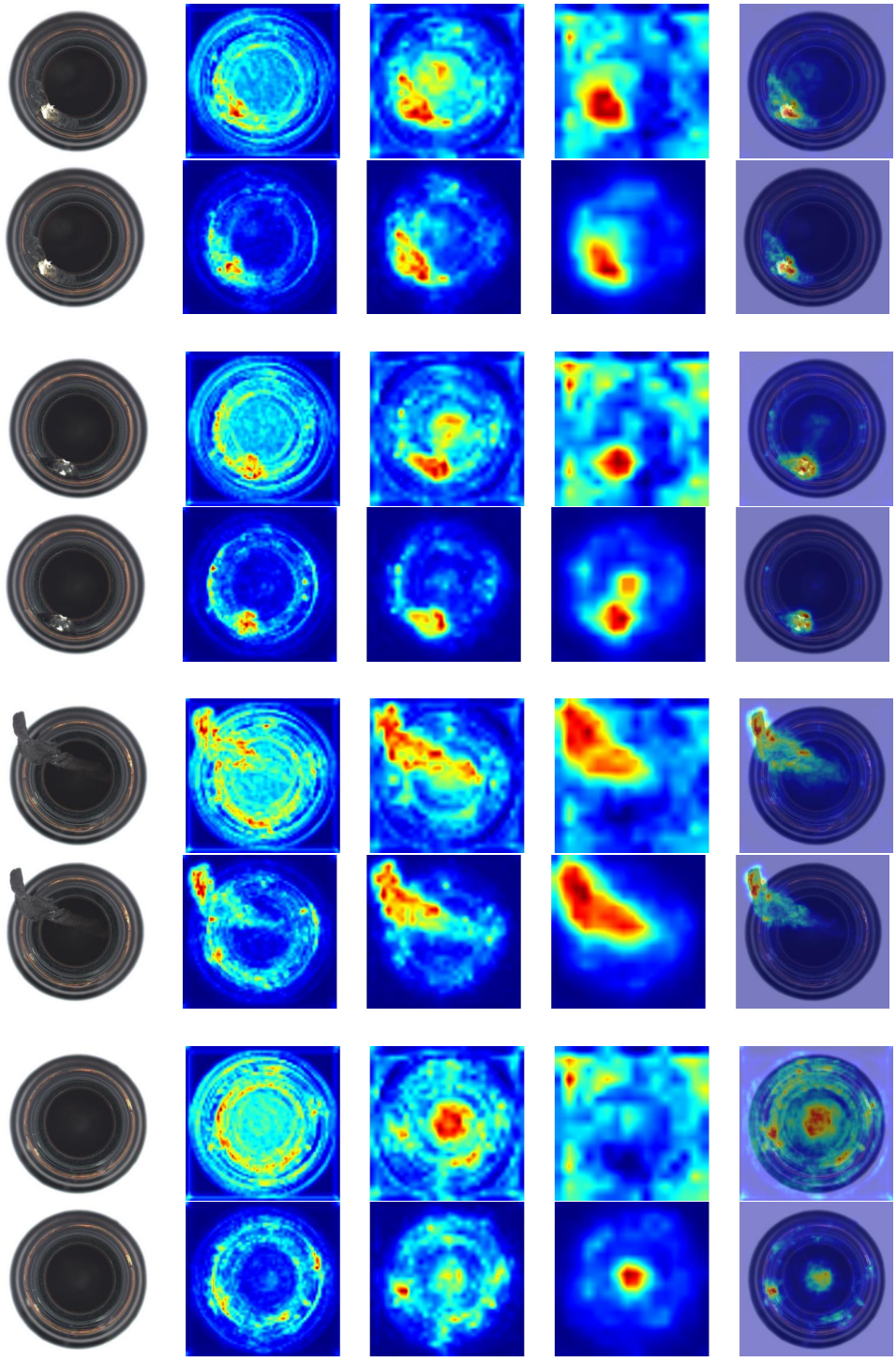

Figure 4: Visual results of our method compared to the classical ST-FPM architecture [2] on four defective images of a bottle. The top row is our model. Columns from left to right correspond to input image, anomaly maps of the three blocks in reverse order (16x16,32x32,64x64) and the resulting anomaly map superposed on the image.

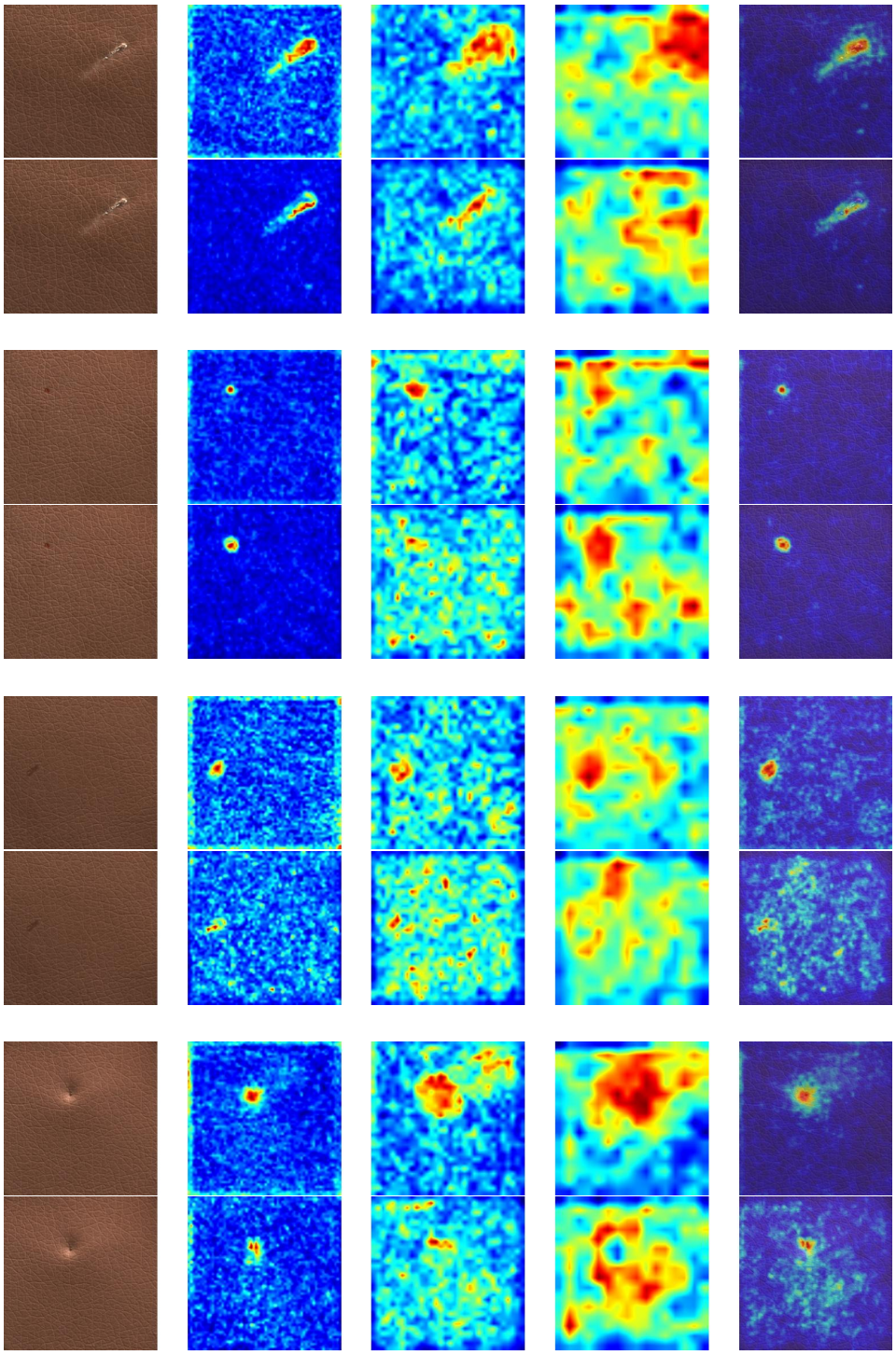

Figure 5: Visual results of our method compared to the classical ST-FPM architecture [2] on four defective images of a leather shred. The top row is our model. Columns from left to right correspond to input image, anomaly maps of the three blocks in reverse order (16x16,32x32,64x64) and the resulting anomaly map superposed on the image.

Table 2: AUC-ROC performance metrics for our model 1 using the custom UNET architecture 2 tested with different hyperparameters on bottles and leather images.

Model	**Pixel Level AUC**	**Original Pixel Level AUC**	**Image Level AUC**	**Epochs**	**Final Training Loss**	**Category**	**Freezed**	**Alpha**
UNET on leather	0.9914	**0.993**	0.9786	100	18.5616	Leather	first 5	1e-5
UNET on leather (001)	0.9903	**0.993**	0.9759	100	18.5175	Leather	train all	1e-5
UNET on leather (002)	0.9292	**0.993**	0.6262	100	29.6038	Leather	first 5	1e-4
UNET on leather (003)	0.9925	**0.993**	0.9976	100	N/A	Leather	first 5	1e-6
UNET on leather (004)	**0.9934**	0.993	0.9973	100	15.759	Leather	first 5	1e-7
UNET on leather (005)	0.9782	**0.993**	0.9823	100	15.4402	Leather	first 5	0
UNET on bottle (001)	**0.9837**	**0.998**	1.000	100	6.3636	Bottle	first 5	1e-7