Industrial Anomaly Detection and Localization on the MVTec Dataset

Davide Vettore
Artificial Intelligence for Science and Technology
University Milano Bicocca
Milan, Italy
d.vettore@campus.unimib.it

***Abstract*—This project explores anomaly detection in industrial settings using the MVTec dataset. Two methods are implemented: a Deep Autoencoder, leveraging reconstruction error, and Multi-Knowledge Distillation, which uses feature-based knowledge transfer. The models are evaluated across five texture categories for anomaly detection and localization. Results show Multi-KD outperforms the Autoencoder in most metrics, particularly for complex anomalies. Detection and localization AUROC metrics are computed, with additional insights from mIoU overlap analysis. The comparison with benchmarks highlight both the advantages and the challenges of the implemented methods. This study provides practical insights into the application of anomaly detection techniques in real-world industrial scenarios.**

***Index Terms*—Anomaly Detection, Anomaly Localization, Autoencoder, Multi-Knowledge Distillation, MVTec AD**

I. Introduction

Anomaly detection refers to the identification of observations that deviate significantly from a learned pattern of regularity, often raising suspicions that these observations were generated by a different mechanism [1]. This task is critical in various applications, such as quality control in manufacturing, medical diagnostics, and video surveillance, where detecting anomalies can prevent costly errors and improve system reliability.

In the context of industrial settings, anomalies like scratches, misalignments, or missing components are uncommon but crucial to detect for maintaining quality and reducing waste. The central challenge in anomaly detection lies in differentiating anomalies from noise within data, as anomalies are typically rare and their characteristics unpredictable. This unsupervised nature makes it necessary to model only the normal data during training, learning a representation that allows anomalies to be identified by their deviation from the normal.

In this project, I explore anomaly detection using state-of-the-art approaches applied to the MVTec dataset [2], a standard benchmark for industrial anomaly detection tasks. To begin, I approach the problem using a Deep Autoencoder, which represents a straightforward yet effective method for detecting anomalies based on reconstruction error. Following this, I implement the more advanced Multi-Knowledge Distillation (Multi-KD) framework [3], a state-of-the-art method that leverages the feature-matching capabilities of deep learning to address the challenges of subtle anomaly localization.

The methodologies chosen for this project are inspired by the comprehensive review presented in *"A Unified Survey on Anomaly, Novelty, Open-Set, and Out- of-Distribution Detection: Solutions and Future Challenges"* [4]. This survey provides a comprehensive overview of state-of-the-art methods across anomaly detection, novelty detection, and related tasks.

II. Dataset

The MVTec Anomaly Detection dataset [2] is a comprehensive benchmark designed for evaluating anomaly detection methods, particularly in industrial inspection scenarios. This dataset includes over 5000 high-resolution images divided into 15 categories, which include five textures (e.g., wood, leather) and ten objects (e.g., metal nuts, pills). The MVTec AD dataset is particularly well-suited for unsupervised anomaly detection because it focuses on detecting small, localized deviations in test images that are not part of the training data. The images were collected under controlled lighting conditions with variations intentionally introduced for certain categories to simulate realistic manufacturing conditions.

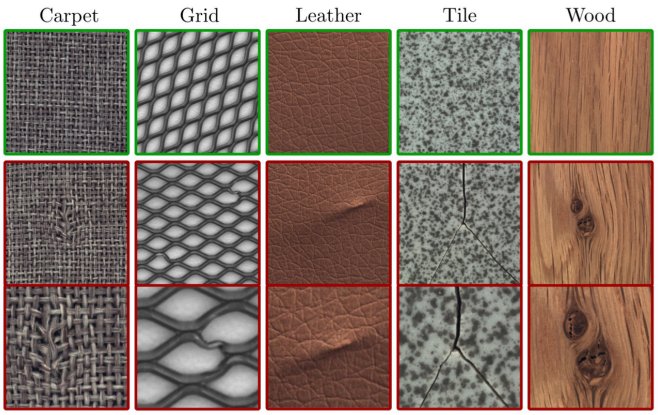

Fig. 1: Example images for all five textures of the MVTec AD dataset. For each texture, the top row shows an anomaly-free image. The middle row shows an anomalous example for which, in the bottom row, a close-up view of the anomalous region is given [2].

In this project, special emphasis was placed on the texture categories within the dataset. These textures, such as carpet, tile, and wood, often exhibit small details and natural variations that test the robustness of anomaly detection methods. The texture categories are illustrated in Fig. 1, providing examples of both anomaly-free and anomalous samples.

III. Methods

This section describes the two approaches used for anomaly detection: first, a Deep Autoencoder based on reconstruction error, and second, the Multi-Knowledge Distillation framework, which leverages feature-based knowledge transfer for anomaly detection.

A. Deep Autoencoder

Deep Autoencoders are neural networks designed to learn efficient representations of input data through encoding and decoding phases. The encoder compresses the input into a lower-dimensional latent space, while the decoder reconstructs the input from this latent representation.

For anomaly detection, the reconstruction error, calculated as the difference between the input and its reconstruction, is used as an indicator of abnormality. High reconstruction errors typically suggest the presence of anomalies, as the Autoencoder is trained only on normal data and struggles to reconstruct inputs deviating significantly from the training distribution.

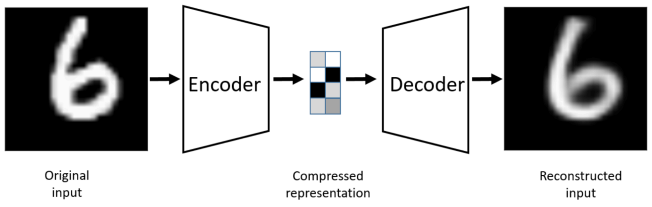

Fig. 2: An Autoencoder example on the MNIST dataset. The input image is encoded to a compressed representation and then decoded [5].

The network architecture employed for this project consists of an encoder, a latent space transformation, and a decoder. The encoder has four convolutional blocks, each comprising a convolutional layer, batch normalization, and a ReLU activation. The latent space is formed through a fully connected layer reducing the feature dimensionality, followed by another fully connected layer to map back into the high-dimensional space required for the decoder. The decoder mirrors the encoder's structure with transposed convolutional layers. To normalize the output to the range $[0, 1]$, a Sigmoid activation is applied at the final layer. This Autoencoder architecture has a total of 13,936,899 trainable parameters, which balances model complexity with the ability to effectively learn features from the training data. Deeper architectures were experimented with but did not yield improvements in anomaly detection performance, suggesting that this configuration is well-suited for the task at hand.

The loss function implemented for training the Deep Autoencoder is the reconstruction loss, which measures how well the model can reconstruct the input data. This loss is calculated using the Mean Squared Error (MSE) between the original input x and its reconstructed output $\tilde{x}$. The mathematical formulation of the reconstruction loss is as follows:

$$\mathcal{L}_{\text{REC}} = \parallel x - \tilde{x} \parallel^2$$

Here, $\parallel x - \tilde{x} \parallel^2$ represents the squared difference between the pixel values of the original input and its reconstruction. By minimizing this loss during training, the Autoencoder learns to accurately encode and decode normal data, ensuring low reconstruction error for such inputs. On the opposite, when an anomalous input is passed through the Autoencoder, the reconstruction error increases, as the network is not trained to represent anomalies.

The Deep Autoencoder was trained using the Adam optimizer with a learning rate of $1 * 10^{-3}$. The training process required only 10 epochs for each texture category, as the model demonstrated quick convergence. Further insights into the training performance and results are discussed in the following section.

B. Multiresolution Knowledge Distillation

Multiresolution Knowledge Distillation is an advanced anomaly detection framework that leverages the principles of knowledge distillation, a branch of transfer learning. This approach utilizes a two-network setup: a robust, pretrained 'expert' network and a simpler 'cloner' network. The expert network extracts rich, multiscale features from the input data, which are then transferred to the cloner by aligning the intermediate outputs of both networks across multiple critical layers. This process ensures that the cloner network learns to replicate the expert's feature representations at different spatial resolutions. During inference, discrepancies between the outputs of the expert and cloner networks at these key points signal anomalies, as the cloner struggles to mimic the expert's feature maps when exposed to anomalous inputs.

Knowledge distillation, as a broader concept, aims to transfer the knowledge of a large, complex model to a smaller, simpler model. This technique enables the deployment of lightweight models that achieve performance similar to their larger counterparts while being more efficient and practical for use in constrained environments, such as edge devices. In knowledge distillation, a loss function is designed to minimize the difference between the outputs of the teacher and student networks. In Multi-KD, instead of relying only on the final output (response-based knowledge) or modeling relationships between multiple outputs (relation-based knowledge), the focus is on aligning intermediate feature activations (Fig. 3). These features are transferable and capture the hierarchical

structure of data representation, making them ideal for tasks like anomaly detection.

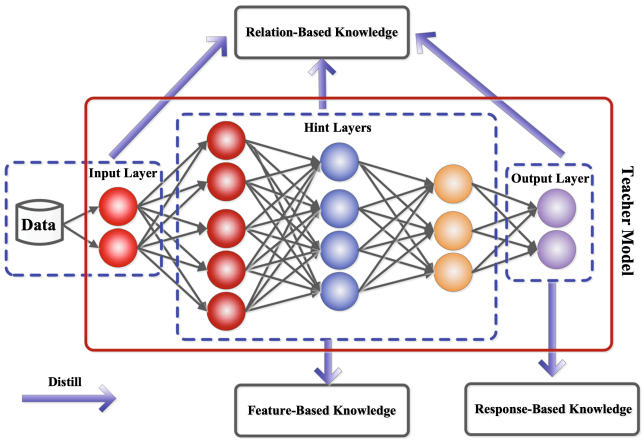

Fig. 3: The schematic illustrations of sources of response-based knowledge, feature-based knowledge and relation-based knowledge in a deep teacher network [6].

This project uses an offline distillation approach, where the expert network is pre-trained and remains fixed while training the cloner. The expert extracts intermediate features that guide the cloner's learning process. A distillation loss ensures that the cloner effectively replicates the expert's representations, despite having a much smaller architecture.

The objective is to reproduce the step-by-step pipeline proposed in the original paper, using similar architectures and theoretical principles to achieve comparable performance.

The expert network utilized is a VGG-16 model pretrained on ImageNet [7], consistent with the architecture employed in the referenced literature. VGG-16 is a deep convolutional neural network known for its simplicity and effectiveness in feature extraction, making it well-suited for knowledge distillation tasks. For this project, only the convolutional and pooling layers of the VGG-16 model are retained. The fully connected layers at the end of the architecture are excluded to focus on feature extraction. The max-pooling layers, which are located at the end of each convolutional block, are selected as the critical points for feature extraction. The input images are resized to $3 \times 128 \times 128$ dimensions, resulting in a model with a total of 14,714,688 trainable parameters.

Fig. 4 shows feature maps extracted by the expert network at the first two critical points, corresponding to max-pooling layers, using an input image of the 'Carpet' class. Each row displays four feature maps out of the total generated by the network. These visualizations demonstrate that earlier layers capture detailed, texture-based features, such as patterns, while deeper layers extract increasingly abstract and high-level features, such as edges and shapes.

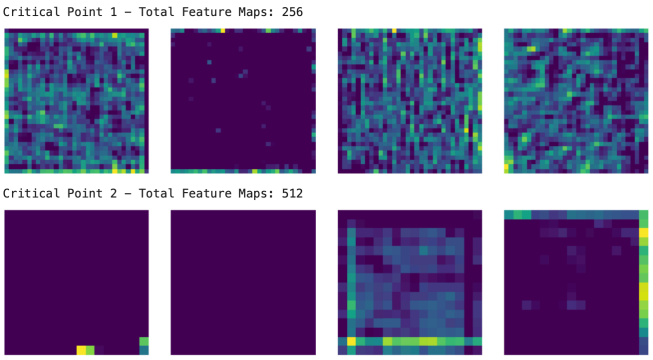

Fig. 4: Visualization of feature maps extracted by the expert VGG16 network at the first two critical points.

The cloner network is a compact version of the expert network, specifically designed to mimic the intermediate features of the VGG-16 at the designated critical points. The architecture used closely resembles the one presented in the literature and contains a total of 9,313,744 trainable parameters. The cloner network is composed of five convolutional blocks, each consisting of two convolutional layers followed by a max-pooling layer. ReLU activation functions are applied after each convolutional layer. This design ensures that the spatial dimensions and feature map sizes at critical points (the max-pooling layers) are aligned with those of the expert network.

The loss function used in the Multiresolution Knowledge Distillation framework consists of two components: the value loss $\mathcal{L}_{val}$ and the directional loss $\mathcal{L}_{dir}$, which together guide the cloner network (C) to replicate the intermediate features of the expert network (S) at predefined critical points. The value loss aims to minimize the Euclidean distance between the activation values of the expert and cloner at each critical layer CP_i. This loss is mathematically defined as:

$$\mathcal{L}_{val} = \sum_{i=1}^{N_{CP}} \frac{1}{N_i} \sum_{j=1}^{N_i} \left(a_s^{CP_i}(j) - a_c^{CP_i}(j)\right)^2$$

Where N_{CP} is the total number of critical points, N_i represents the number of neurons in the i-th critical layer CP_i and $a_s^{CP_i}(j)$ and $a_c^{CP_i}(j)$ denote the j-th activation values of the expert and cloner networks, respectively, at critical point CP_i. This loss ensures that the numerical values of activations in the cloner closely match those of the expert network. The directional loss, on the other hand, is designed to increase the directional similarity between the activation vectors of the expert and cloner networks. This is especially important for ReLU-based networks, where directional alignment affects how subsequent layers process the features. To force this similarity, the directional loss employs the cosine similarity metric, defined as:

$$\mathcal{L}_{dir} = \sum_i \left(1 - \frac{\text{vec}(a_s^{CP_i})^T \cdot \text{vec}(a_c^{CP_i})}{\| \text{vec}(a_s^{CP_i}) \| \, \| \text{vec}(a_c^{CP_i}) \|}\right)$$

In this formulation, $\text{vec}(a_s^{CP_i})$ and $\text{vec}(a_c^{CP_i})$ are the flattened

activation vectors of the expert and cloner networks at critical point CP_i. The numerator represents the dot product of the two vectors, while the denominator normalizes the result by their magnitudes to measure their alignment. The directional loss complements the value loss by aligning not just the magnitudes but also the directions of the feature vectors. The total loss combines these two components as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{val}} + \lambda \mathcal{L}_{\text{dir}}$$

λ is a hyperparameter that controls the trade-off between the value and directional losses. Although the paper proposes methods for tuning, experiments revealed that lower values of λ perform better on the MVTec AD dataset.

The training process employs the same Adam optimizer and learning rate ($1 * 10^{-3}$) used in the previous approach. However, due to the increased complexity of this method, the number of training epochs is set to 75 for each category. Some categories, however, showed delayed convergence and required extended training periods over later experiments to achieve optimal performance.

IV. Results

In this section, I evaluate the performance of the two implemented approaches applied across five different texture categories. The evaluation is divided into two main parts: anomaly detection and anomaly localization. For anomaly detection, I compute anomaly scores, analyze detection AUROC as the primary metric, and assess additional metrics such as accuracy and precision. For anomaly localization, I compare the spatial identification of anomalies using localization AUROC, visualize examples, and calculate mean intersection over union (mIoU) as an overlap metric with ground-truth masks.

A. Anomaly Detection

For the Deep Autoencoder approach, anomaly scores are computed as reconstruction errors, using the same metric applied during training. Specifically, the reconstruction error for each test image is calculated as the mean squared difference between the input image and its reconstructed version. These raw anomaly scores are then normalized to the interval $[0, 1]$ for consistency.

For the Multi-Knowledge Distillation framework, anomaly scores are derived by calculating the discrepancies between the outputs of the expert and cloner networks at each critical point. The discrepancy for a given critical point is computed as the mean squared difference between the activation maps of the expert and cloner. These discrepancies are aggregated across all critical points to produce the final anomaly score for each image, which is also normalized to the range $[0, 1]$.

Using these anomaly scores, the primary metric for evaluation is the detection AUROC (Area Under the Receiver Operating Characteristic Curve). This metric quantifies the model's ability to distinguish between normal and anomalous samples without requiring a threshold. A higher AUROC indicates better discrimination between the two classes. In Fig. 5, the detection ROC curves obtained for both methods across the five texture categories are shown, providing a visual representation of their performance.

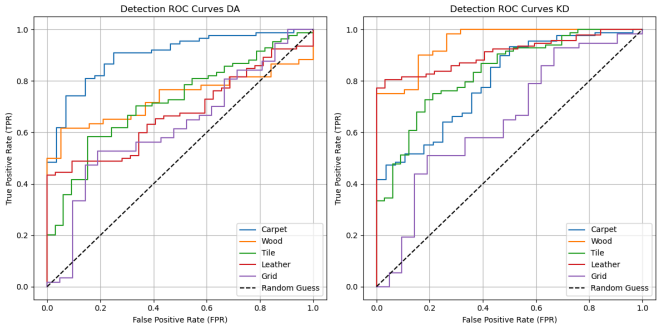

Fig. 5: Detection ROC curves obtained for both methods across the five texture categories.

In addition, thresholds were established based on the frequency distribution of anomaly scores or using specific percentiles. These thresholds allowed the computation of metrics such as accuracy and precision. Accuracy measures the overall correctness of predictions, while precision focuses on the proportion of true anomalies among all samples classified as anomalous. Precision is particularly important in scenarios where the primary goal is to reliably detect anomalies.

TABLE I: Detection AUROC achieved by the two models across the five texture categories in %.

Method	Carpet	Grid	Leather	Tile	Wood
DA	90.09	62.32	67.60	72.19	73.60
KD	79.90	64.66	90.79	83.26	95.09

Table I shows the detection AUROC achieved by the two models across the five texture categories. As observed, the second approach (KD) generally outperforms the simpler autoencoder-based approach (DA) across most categories, with the exception of "Carpet". This category appears particularly challenging for the Multi-KD framework during the detection phase. The superior performance of DA in "Carpet" is confirmed by its higher accuracy (86.32% vs. 74.36%) and precision (91.95% vs. 85.54%) in detecting anomalies. For other textures, Multi-KD consistently demonstrates better performance, confirming its effectiveness. "Grid," however, emerges as a challenging class for both models, achieving relatively low AUROC values.

To visually illustrate what the two models consider anomalous, Fig. 6 and Fig. 7 display the five images with the lowest and highest anomaly scores for the "Carpet" category, as determined by the Deep Autoencoder and Multi-Knowledge Distillation models, respectively.

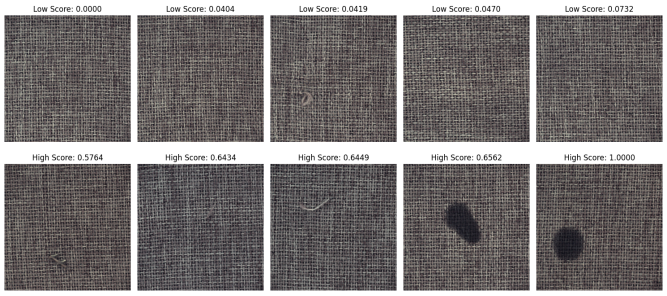

Fig. 6: Images with the lowest and highest anomaly scores for the "Carpet" category, as determined by the Deep Autoencoder model.

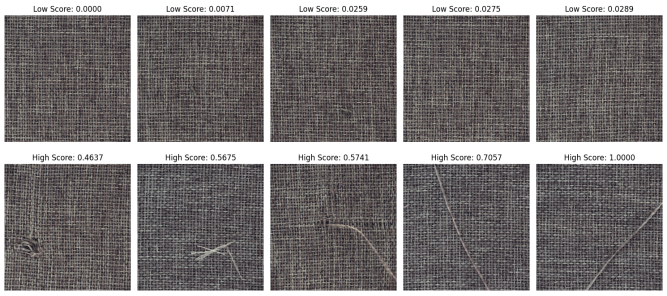

Fig. 7: Images with the lowest and highest anomaly scores for the "Carpet"category, as determined by the Multi-Knowledge Distillation model.

B. Anomaly Localization

In the Deep Autoencoder approach, the anomaly localization process starts with the reconstruction error, which was previously averaged across all pixels to compute a single anomaly score for each image. For localization, this reconstruction error is retained at the pixel level, where the error is averaged across the color channels to obtain an anomaly score for each pixel. The resulting heatmap highlights regions in the image where reconstruction errors are significant. To refine the raw heatmap and localize anomalies more effectively, a series of processing steps are applied, inspired by the pipeline proposed in the Multi-KD literature [3]. First, the raw heatmap is normalized to the range $[0, 1]$. A percentile-based thresholding method is then applied, retaining only the top 2% of pixel values, which are likely to correspond to anomalous regions. To further reduce noise and enhance the map, the processed heatmap undergoes Gaussian blurring, followed by morphological operations. An opening filter with an elliptical structuring element is applied to eliminate noise and smooth the heatmap for better clarity.

In the Multi-Knowledge Distillation framework, anomaly localization relies on the gradients of the total loss function with respect to the input image. These gradients are proven to capture the contribution of each pixel to the overall loss value, providing meaningful insights into the significance of each pixel in determining the anomaly. To generate the raw attribution map, the input image is treated as a variable requiring gradients. The total loss between the expert and cloner networks is computed, and the gradients of this loss with respect to the input image are obtained through backpropagation. The resulting gradients are then aggregated across the color channels, using the sum of their absolute values to highlight the most impactful pixels. This produces a raw attribution map where higher values correspond to regions contributing most significantly to the anomaly score. The raw attribution map is then processed using the same pipeline as in the Deep Autoencoder approach. This includes normalization, percentile-based thresholding, Gaussian blurring, and morphological filtering to produce a refined localization map.

The processed heatmaps can be compared against the binary ground truth masks to evaluate the models' localization performance. This comparison allows for the computation of the Localization AUROC, which measures how well the models distinguish between anomalous and non-anomalous regions. The ROC curves for the localization results of the two models across the five texture categories are shown in Fig. 8.

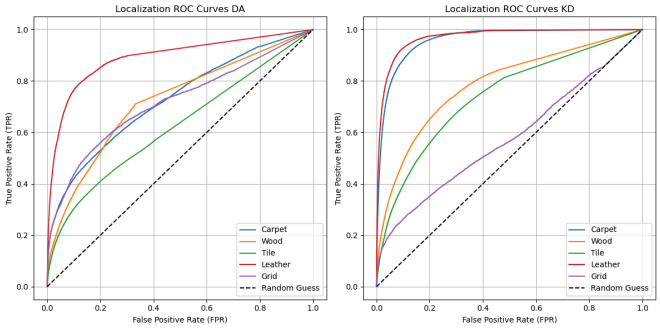

Fig. 8: Localization ROC curves obtained for both methods across the five texture categories.

In Table II, the Localization AUROC achieved by the two models on the five texture categories is summarized. The Multi-KD framework outperforms the autoencoder-based approach in all categories except "Grid." However, further experiments revealed that the "Grid" category shows slower convergence and required extended training periods to achieve optimal performance. For instance, by doubling the training epochs, the Multi-KD model was able to reach a localization AUROC of 72.03% on the "Grid" category. Another interesting observation is the performance on the "Carpet" category, despite showing poor results during the detection phase, the Multi-KD framework significantly outperforms the simpler Autoencoder approach here, achieving one of the highest localization AUROC across all categories.

TABLE II: Localization AUROC achieved by the two models across the five texture categories in %.

Method	Carpet	Grid	Leather	Tile	Wood
DA	72.59	71.68	89.15	62.35	72.15
KD	95.93	57.72	97.02	74.14	78.64

Additionally, Fig. 9 and Fig. 10 provide visual examples of anomalies localized by the two methods on two different categories. Fig. 9 shows an example from the "Carpet" category,

which exhibits the greatest difference in localization AUROC between the two models. While the Multi-KD model almost flawlessly identifies the anomaly, the autoencoder-based approach struggles and focuses on noise instead, as is particularly evident in the raw anomaly maps. In contrast, Fig. 10 shows an example from the "Leather" category, where the two models achieved comparable localization performance. In this sample, the anomaly is accurately detected by both models, and the Autoencoder even appears to highlight the cut with slightly more precision in this specific instance.

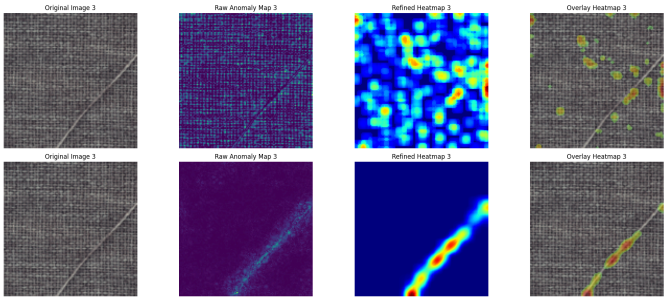

Fig. 9: Anomaly localization on the "Carpet" category. The top row illustrates localization performed by the autoencoder-based framework, while the bottom row shows results from the Multi-Knowledge Distillation model.

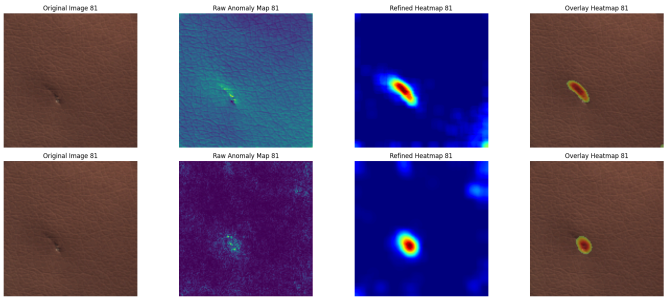

Fig. 10: Anomaly localization on the "Leather" category. The top row illustrates localization performed by the autoencoder-based framework, while the bottom row shows results from the Multi-Knowledge Distillation model.

Overlap Metric:

To further evaluate anomaly localization, I introduced a metric that, unlike AUROC, requires thresholding the heatmaps, similar to the accuracy and precision metrics used in the detection phase. Specifically, I compute the mean Intersection over Union (mIoU) between the binary ground truth mask and the thresholded anomaly heatmaps. This metric measures the degree of overlap between the predicted anomalous regions and the ground truth, providing a clear indication of how accurately the model localizes anomalies.

Table III shows the mean Intersection over Union (mIoU) achieved by the two models across the five texture categories. The overall low values highlight that, as expected, the unsupervised nature of the problem limits the anomaly segmentation performance. However, the results clearly demonstrate that the Multi-KD framework outperforms the simpler Autoencoder model in anomaly localization across all categories except "Grid."

TABLE III: Mean IoU achieved by the two models across the five texture categories in %.

Method	Carpet	Grid	Leather	Tile	Wood
DA	14.5	8.9	18.5	9.8	12.6
KD	34.4	6	27.2	14.2	21.7

Fig. 11 shows samples of binary masks predicted by the two models, along with the corresponding ground truth masks, for the "Carpet" category. In categories with higher overall mIoU, some masks are visually close to the ground truth. In the first two rows, the masks generated by the Multi-KD model are clearly better aligned with the ground truth, as reflected by the higher IoU, Dice, and pixel-wise accuracy metrics. However, in the third row, the Autoencoder approach predicts a binary mask that more closely matches the ground truth, demonstrating its occasional superiority in specific samples.

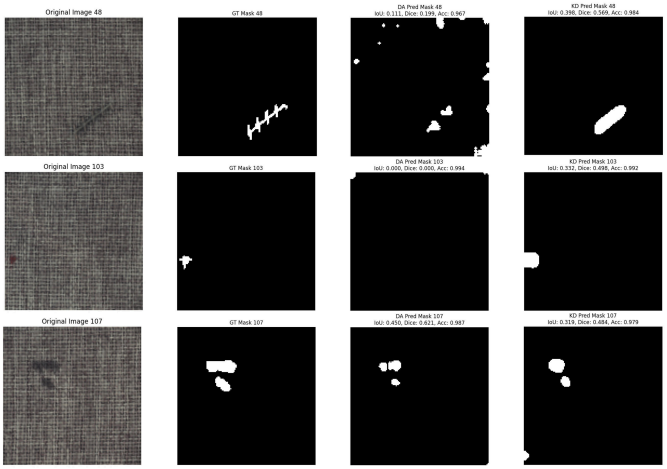

Fig. 11: Examples of binary masks predicted by the Autoencoder and Multi-KD models, alongside the ground truth, for the "Carpet" category. Metrics such as IoU, Dice, and accuracy are provided for each prediction.

V. Conclusions

This study compared two approaches for anomaly detection and localization on the MVTec dataset: a Deep Autoencoder and the Multi-Knowledge Distillation framework. The Autoencoder, a simpler reconstruction-based method, offered reasonable performance in detection and localization but struggled with more complex anomalies. On the other hand, Multi-KD performed better on most metrics, showing its ability to detect and localize anomalies more accurately by using feature differences.

Comparing the results of the implemented Multi-KD model with those reported in the original paper, I observe comparable performance across most texture categories. In the anomaly detection task, my model slightly outperforms the reference AUROC in the "Carpet" and "Wood" categories. However, it falls slightly short in the "Grid" and "Tile" categories, which I

identified as benefiting from additional training. Similar trends are observed in the localization task, where three out of five categories achieve nearly identical results, while the same two underperforming categories highlight the need for extended training or reconsideration in future experiments.

When testing the two models implemented in this project across all classes of the MVTec Dataset, I achieved mean detection AUROCs of 60.64% and 75.48%, respectively. Comparing these results to the anomaly detection benchmark proposed by M. Salehi in [4], the Deep Autoencoder model falls short of the top-performing models, while my Multi-KD implementation ranks just outside the podium. It's important to note that my focus was on texture categories, and when extending to all categories, each model was trained for only 75 epochs. Some slower-converging categories may have benefitted from longer training times, as I've already observed while studying the texture categories. Additionally, the "Metal nut" category performed particularly poorly, with a detection AUROC of just 29.86%, significantly lowering the overall average and suggesting further investigation.

For anomaly localization, the models achieved mean AUROCs of 61.73% and 74.47%, respectively, which are less competitive compared to the 90.71% reported in the reference paper. However, as with detection, the models were originally designed and optimized for texture categories and were applied to the entire dataset without adjustments or extended training times. Specific categories, such as "Metal nut" and "Pill" showed significantly lower performance (40.12% and 52.93% with the Multi-KD model), dragging down the overall results and highlighting key weaknesses of the models.

Future experiments could focus on exploring different expert-cloner model architectures within the knowledge distillation framework. Beyond knowledge distillation, other state-of-the-art frameworks for anomaly detection could be investigated. Approaches such as GAN-based methods, self-supervised contrastive learning, or transformer-based models might provide new insights and push performance further.

Overall, this project focused on anomaly detection in industrial settings and demonstrated the effectiveness of two different approaches on the MVTec dataset. The results highlighted the strengths and weaknesses of reconstruction-based and knowledge-distillation-based methods, providing useful insights into their practical use.

References

[1] D. Hawkins, "Identification of outliers." Chapman, Hall, 1980.

[2] P. Bergmann, M. Fauser, D. Sattlegger, and C. 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*, 2019, pp. 9592–9600.

[3] M. Salehi, N. Sadjadi, S. Baselizadeh, M. H. Rohban, and H. R. Rabiee, "Multiresolution knowledge distillation for anomaly detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 14902–14912.

[4] M. Salehi, H. Mirzaei, D. Hendrycks, Y. Li, M. H. Rohban, and M. Sabokrou, "A unified survey on anomaly, novelty, open-set, and out-of-distribution detection: Solutions and future challenges," *arXiv preprint arXiv:2110.14051*, 2021.

[5] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," *Machine learning for data science handbook: data mining and knowledge discovery handbook*, pp. 353–374, 2023.

[6] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789–1819, 2021.

[7] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition." [Online]. Available: https://arxiv.org/abs/1409.1556