

ZERO & FEW-SHOT ANOMALY DETECTION USING DINOv3

William Peytz (s204145), Rian Gonzenbach (s251763), Alexander Wassmer (s253740)

ABSTRACT

This project investigates the use of self-supervised vision foundation models, specifically DINOv3, for industrial anomaly detection with minimal task-specific training. We evaluate zero-shot, few-shot, mirroring, and prompt-based feature adaptation strategies across five diverse MVTec AD categories. Zero-shot detection performs well on objects with consistent appearance but struggles on categories with high variance or small defects. Few-shot learning and prompt-based adaptation improve performance, with the latter achieving the highest AUROC gains by applying targeted per-patch embedding adjustments. Additionally, we use SAM to segment anomalous regions, which works reliably for objects with clear boundaries but requires finetuning for low-contrast or diffuse defects. Overall, our results show that combinations of foundation models with adaptation and segmentation methods are feasible to use with proper finetuning for a specific category or product and can reduce the need for labeled data.

1. INTRODUCTION

Industrial quality control has traditionally relied on rule-based computer vision systems or supervised machine learning models that require extensive labeled defect datasets for training. This procedure faces significant challenges: collecting sufficient defect samples is costly and timeconsuming, defect patterns vary across product types, and models require retraining when new defect categories emerge.

Recent advances in self-supervised vision foundation models, such as DINOv3 [1], present an alternative. Trained on millions of diverse images without human annotations, these models learn visual representations that capture patterns of normality and deviation. They can perform anomaly detection in a manner, identifying defects without ever or only seeing few samples during training, by comparing test samples against learned representations of normal patterns.

This project investigates whether vision foundation models can achieve competitive anomaly detection performance without task-specific training. We look at if a zero-shot method can be a good baseline for anomaly detection. In order to improve AUROC scores we also look at other methods such as a few-shot method, as well as mirroring the DINO model and finally also a prompt based feature adaptation for

anomaly detection.

2. METHODS

2.1. MVTec Anomaly Detection Dataset

The MVTec Anomaly Detection (MVTec AD) dataset, introduced by Bergmann et al. in 2019 [2], has become the standard dataset for industrial defect detection research. It includes 15 product categories spanning textures and objects. Each category contains:

- **Training set:** 200-400 defect-free samples.
- **Test set:** Classified normal and defective samples with various defect types.
- **Ground truth masks:** Pixel-level annotations for defect localization evaluation

It was decided to focus on five diverse categories to evaluate performance across different structural characteristics: *Bottle* (symmetric), *Carpet* (texture), *Cable* (asymmetric), *Screw* (high variance), *Hazelnut* (natural variance). Figure 1 shows these categories. Additional examples of normal and defects are illustrated in A.1.



(a) Hazelnut (b) Carpet (c) Bottle (d) Screw (e) Cable

Fig. 1: Normal samples of the analyzed categories.

2.2. Similarity Metrics

To quantify the similarity of patches within the image for zero-shot and between test samples and normal samples for few-shot detection, three metrics are considered:

- **Cosine similarity:** angular similarity between embeddings x and y (Eq. 1).
- **L2 distance:** Euclidean distance between embeddings x and y (Eq. 2).
- **k-NN distance:** Average distance from a test embedding x to its k nearest neighbor embeddings $\{y_1, \dots, y_k\}$ (Eq. 3).

2.3. Zero-Shot and Few-Shot Anomaly Detection

As shown in the third panel of Figure A.2, zero-shot detection relies on internal consistency without external reference images. The test image is passed through the frozen DINOv3 backbone to extract a 14x14 grid of patch embeddings. An anomaly score is calculated by comparing individual patches to the mean embedding of the same image.

The second panel of Figure A.2 illustrates that for few-shot detection reference embeddings are extracted from normal training images using the DINOv3 model. During detection on test images, patches from the test image are matched with patches of the normal samples. For the matching patches, one of the similarity metrics described in Section 2.2 is used to obtain an anomaly score. Patches dissimilar to normal patch-embeddings receive a higher anomaly score. The patch level similarity scores are then used to calculate an image-level anomaly score using the mean of all patches of the image.

2.4. Use of SAM

In this project, SAM (Segment Anything Model) [3] is applied to obtain a pixel-level mask of the defect area based on the 5-shot anomaly map generated by DINOv3. The inputs are the original image of the object, along with positive and negative point prompts. Positive points are randomly sampled within regions where the anomaly score is $\geq 90\%$, while negative points are placed in the remaining areas of the image to prevent SAM from segmenting overly large regions.

2.5. Mirroring DINOv3 Model

A *mirroring* model, shown in the first panel of Figure A.2, replaces reference embeddings (few-shot) with a reference image generated by horizontally flipping the test image. Embeddings are extracted from both the original and flipped images; the flipped embeddings are compared directly to the original embeddings to enforce horizontal symmetry. The resulting symmetry error is used as the anomaly score. This method should be effective for the symmetric objects like *bottles*, and unsuitable for asymmetric objects like *cables* and *hazelnuts*.

2.6. Use of prompt based feature adaptation

To adapt the frozen DINOv3 backbone to domain-specific anomaly detection without full fine-tuning, a small set of learnable prompt vectors that guide feature modulation are introduced. Each patch embedding attends to these prompts and is transformed by a multilayer perceptron, MLP, producing a per-patch modulation that is added to the original features. The prompts and MLP are trained using a separability loss that encourages normal samples to cluster tightly while pushing defect samples away. Training requires approximately 30 defect samples per category and completes in under 2 minutes, making this a practical adaptation strategy.

3. RESULTS

3.1. Similarity Metrics

The three different similarity metrics described in Section 2.2 were tested against all categories. Table 1 shows the mean performance for the classification into normal and defect objects across the five categories for 5-shot anomaly detection. All three similarity metrics show very similar scores, with K-NN having an edge in AUROC, Recall and F1 and is therefore used for the further evaluations.

Metric	AUROC	Precision	Recall	F1
Cosine	0.789	0.822	0.959	0.880
Euclidean	0.786	0.8327	0.931	0.874
k-NN	0.799	0.838	0.937	0.880

Table 1: Mean performance metrics over the categories for different similarity metrics

3.2. Effect of number of normal reference images for Few-shot

To evaluate how the number of normal reference images affect classification, the AUROC across categories was calculated using the k-NN similarity metric with $k=5$. The results are shown in Figure 2.

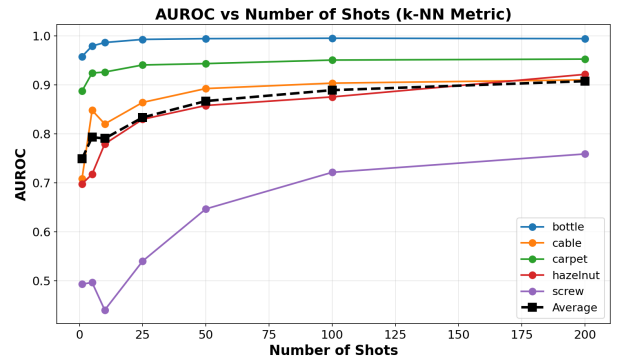


Fig. 2: AUROC score for the categories depending on the number of normal training samples provided.

3.3. Anomaly Maps of different models

Figure 3 compares the anomaly maps obtained with the different models analyzed in this report, using defect instances of the *Carpet* and the *Hazelnut*. The results are discussed in Section 4.3.

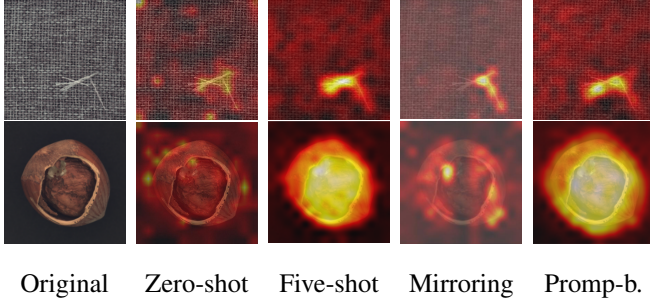


Fig. 3: Original image and anomaly maps generated by different models for two defect objects.

3.4. Fewshot Anomaly Segmentation

Figure 3 shows the original image, computed anomaly map, and resulting SAM mask overlayed on the ground truth mask of two objects. Table 2 shows the quantitative results of the SAM implementation for the assessed categories. These are discussed in Section 4.4.

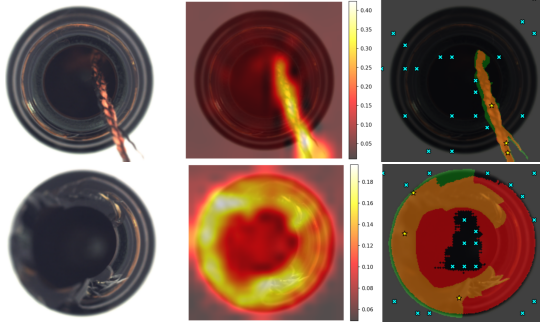


Fig. 4: Good (top) and bad (bottom) example of SAM segmentation with original image on the left, anomaly map in the middle and segmentation on the right, where green is the groundtruth and red the predicted defect area.

Table 2: Segmentation metrics for five categories

Category	Dice	IoU	Recall	Precision
Cable	0.61	0.44	0.80	0.50
Bottle	0.59	0.42	0.69	0.51
Hazelnut	0.39	0.24	0.86	0.25
Carpet	0.18	0.10	0.57	0.11

3.5. Method Ablation Study of Models

In Table 3, the Prompt-based achieves the highest performance across all categories, most notably in the high variance ‘screw’ category (0.984) where other methods fail. The Mirroring method is fairly competitive on symmetric objects

(‘bottle’) and textures (‘carpet’) but, as hypothesized, fails on the asymmetric ‘cable’ (0.554). Few-shot provides a strong baseline for consistent objects but struggles with high variance (‘screw’, 0.483).

Table 3: Cosine AUROC performance scores comparison across diverse categories.

Category	Zero	Few (n=5)	Mirroring	Prompt + 5-shot	Prompt (full)
Bottle	0.7937	0.9849	0.9802	0.9849	0.992
Carpet	0.6200	0.9318	0.9097	0.9318	0.960
Cable	0.6089	0.8126	0.7397	0.8126	0.932
Screw	0.5325	0.4829	0.6024	0.4839	0.773
Hazelnut	0.5204	0.7332	0.4893	0.7336	0.928

3.6. Prompt based feature adaptation

We investigated whether learned visual prompts could improve anomaly detection by adapting frozen DINOv3 features. Our per-patch learned transform approach achieves substantial improvements when using all training samples as reference, with an average AUROC increase of 30.2% across five categories. The most challenging category (hazelnut) benefits the most, improving from 0.5204 to 0.928 (+40.8%). However, when evaluated in a 5-shot setting, prompts provide negligible improvement over the baseline few-shot method (Table 3), as the learned feature transformation requires a sufficient reference set to be effective. This result suggests that prompt-based adaptation is best suited for scenarios where normal reference samples are abundant but labeled defect examples are scarce.

4. DISCUSSION

4.1. Similarity metrics

In Section 3.1, Cosine similarity, Euclidean distance, and k -NN ($k = 5$) showed nearly identical performance. This result is expected as DINOv3 produces well separated and compact embeddings for normal and abnormal samples, making the choice of similarity metric less impactful. Further, DINOv3 applies L2-normalization to its embeddings. For normalized vectors x and y with $\|x\| = \|y\| = 1$, the Euclidean distance simplifies to

$$\|x - y\|^2 = 2(1 - \cos \theta),$$

showing that Euclidean distance in this case is a transformation of Cosine similarity. Therefore, both metrics, and consequently k -NN, behave almost identically in this feature space.

4.2. Classification Performance Across Categories

As shown in Figure A.3, the few-shot classification results vary considerably. *Bottle* and *Carpet* achieve the highest AUROC due to consistent geometry and repetitive textures, making defects clearly distinguishable. Performance drops for

Cable ($\approx 15\%$) and *Hazelnut* due to greater natural variance in shape and viewpoint. The lowest AUROC is observed for *screw* ($\approx 50\%$), where fine details and texture variations make subtle defects hard to capture with 14×14 patches. This patch size may be insufficient to capture such small anomalies, and additional factors such as low contrast and lighting variation further reduce performance. An inspection of the error types in Table A.2 show that this results in poor Recall (0.45 for zero-shot, 0.71 for few-shot). This indicates a high number of False Negatives (an essential metric for industrial adoption where missing a defect is more significant than a False Positive).

While increasing the number of shots consistently improves performance (Figure 2), the gain for *Bottle* and *Carpet* is marginal because their normal samples have smaller variance, so additional shots contribute limited new information. In contrast, categories like *Hazelnut* and especially *Screw* benefit noticeably from more samples, as their variation in texture, geometry, and viewpoint require a greater number of examples to generalize well.

4.3. Anomaly Maps for different Models

Figure 3 presents the anomaly maps of the different models for the *Carpet* and the texture categories. These two examples were selected because they are among the most representative of all categories. It is evident that the zero-shot anomaly map fails to recognize the defect in the *Hazelnut* example. This is reasonable, as the patches within a hazelnut all vary in color and shape, making it difficult for the model to generalize what a normal patch should look like. In contrast, the normal *Carpet* images have a uniform texture across its entire surface, enabling the model to compare patches and reliably detect defects. The anomaly map of the prompt-based model, trained on all available training images, shows even more refined results. The five-shot model performs significantly better for both objects, as it is provided with five training samples from which it can learn the appearance of a normal object. Anomalies are successfully detected with high contrast relative to the background. However, for the hazelnut, the area surrounding the crack is also marked as anomalous which is an understandable outcome given the unique and variable texture of each hazelnut. The mirroring model successfully detects most defects in the carpet. This is expected, since the method identifies anomalies by comparing the left and right halves of the image. As a consequence, the model fails to detect the hazelnut defect, as this approach is not suitable for objects without horizontal symmetry.

4.4. SAM Segmentation

As shown in the top row of Figure 4, the predicted defect region (red) aligns closely with the ground truth (green). This strong performance comes from two factors: the anomaly map places positive prompts accurately on true hotspots, and

the string’s color provides strong contrast against the black bottle, giving SAM a clear boundary to segment. In the lower example, however, SAM predicts a larger defect area due to the weak contrast and gradual transition between defective and intact regions.

Table 2 reflects this trend: the cable category performs best because of its sharp edges, whereas carpet scores much lower due to its diffuse, transitional boundaries. Potential improvements include tuning the point-segmentation threshold or adjusting the number of positive and negative points per category.

4.5. Prompt based feature adaptation

The key insight enabling these results is that effective feature adaptation requires per-patch modulation rather than image-level adjustment. The initial implementation applied identical offsets to all patches, which preserved relative Cosine similarities and yielded no improvement. By allowing each patch to attend to prompts independently, the model can selectively enhance features important for anomaly detection. The number of defect samples also proved critical. With only 5 samples (few-shot setting), the model cannot learn generalizable feature adaptations, resulting in negligible improvement over the baseline (few-shot).

5. CONCLUSION

This project shows that self-supervised vision models like DINOv3 can be applied to industrial anomaly detection with very little task-specific training. Zero-shot detection gives good results for categories with stable appearance and clear structure, but performs poorly when the objects have high natural variation or when defects are very small. Adding a few normal reference images noticeably improves performance. In general, the more training samples improve the results. The mirroring approach behaves as expected: it works well for symmetric and textured categories but fails on asymmetric ones because left-right comparison is not meaningful. The prompt-based feature adaptation method gives further improvement. It significantly increases AUROC scores, showing that small, targeted changes to the feature space are enough to adapt the model to more difficult categories. SAM segmentation generally works well when the anomaly map highlights the defect clearly and when the object has sharp boundaries. However, segmentation becomes unreliable for categories where defects blend smoothly into the background, such as carpets. Further, finetuning on a specific objects is necessary for reliable results. Overall, the results indicate that foundation models like DINOv3 combined with adaptation and finetuning for specific objects can provide an interesting alternative to conventional anomaly detection.

6. REFERENCES

- [1] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khali-dov, Marc Szafraniec, Seungeun Yi, Michaël Ramamon-jisoa, Francisco Massa, Daniel Haziza, Luca Wehrst-edt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski, “DI-NOv3,” 2025.
- [2] 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 (CVPR)*, 2019, pp. 9592–9600.
- [3] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Pi-otr Dollár, and Ross Girshick, “Segment anything,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 4015–4026.

A. ADDITIONAL MVTEC AD DATASET EXAMPLES

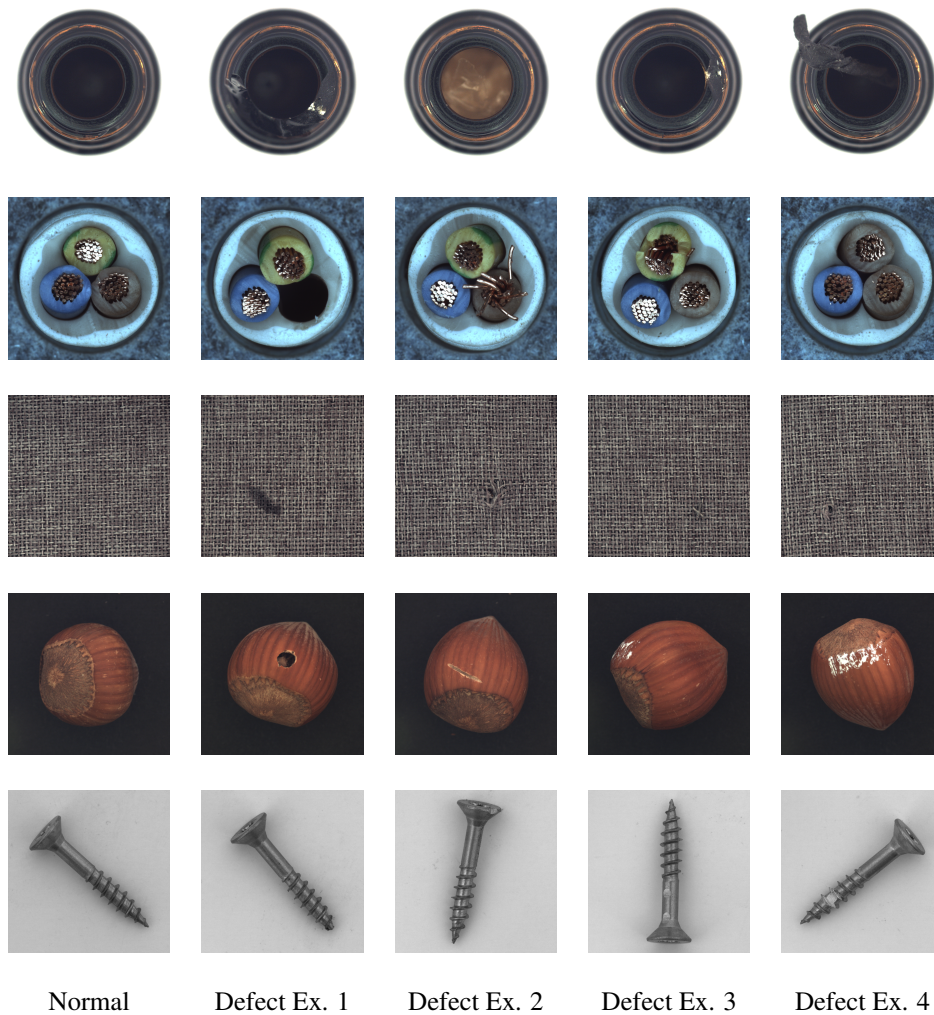


Fig. A.1: Instances of normal and defective samples of the five chosen categories across five rows. Each column represents a type of sample.

B. ARCHITECTURES OF METHODS

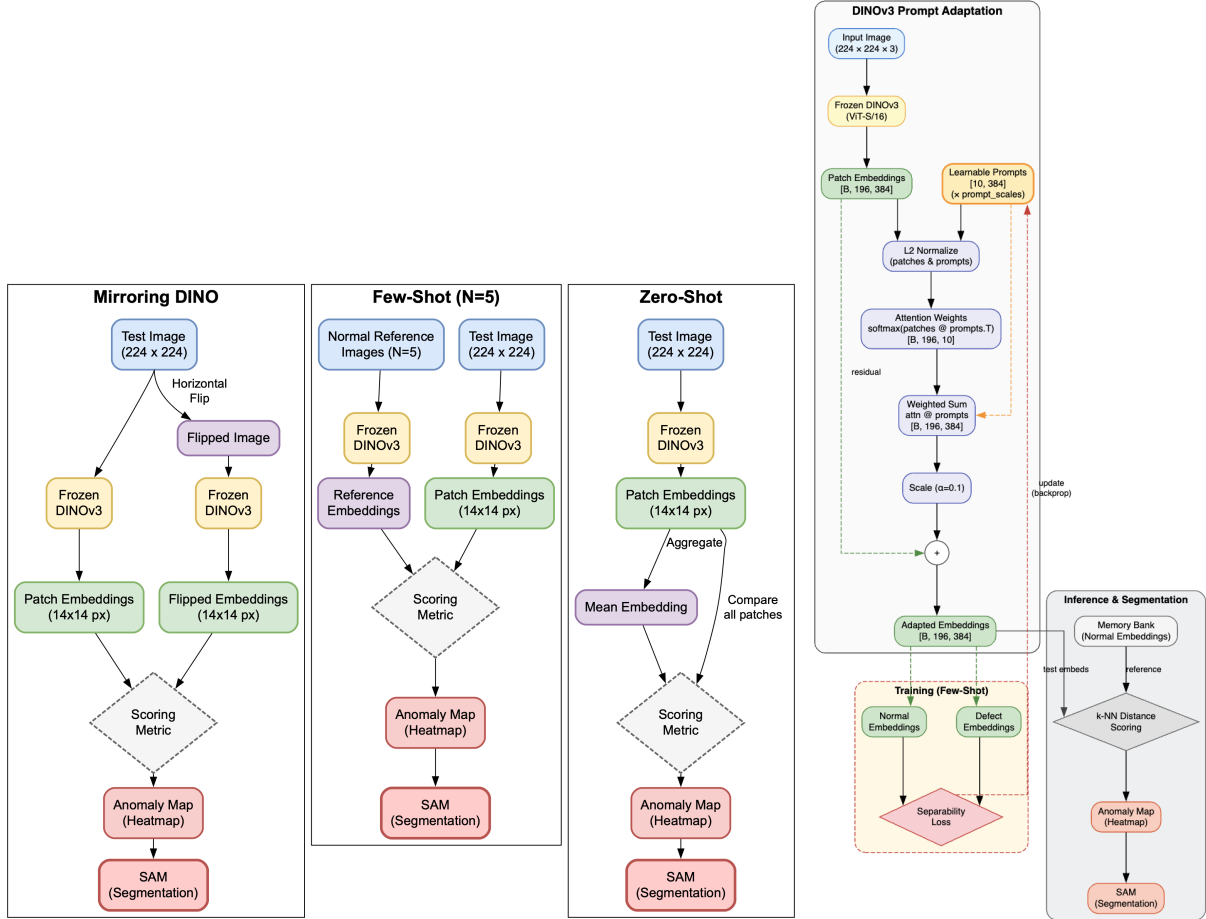


Fig. A.2: Architectural comparison of anomaly detection strategies: **(Left) Mirroring** generates a reference image by horizontally flipping the test image, exploiting geometric symmetry to identify defects. **(Middle Left) Few-Shot** compares features against reference embeddings of $N = 5$ normal reference images using Cosine, Euclidean, or k-NN metrics. **(Middle Right) Zero-Shot** uses internal consistency, comparing patches to the intra-image average to compute an anomaly score ($1 - \text{Similarity}$). Finally, SAM is applied to the heatmaps to generate segmentation masks.

C. AUROC SCORE FOR 5 CATEGORIES FOR DIFFERENT SIMILARITY METRICS

Figure A.3 demonstrates that the cosine, euclidian and k-nn similarity metric provide almost identical results for the 5 different categories.

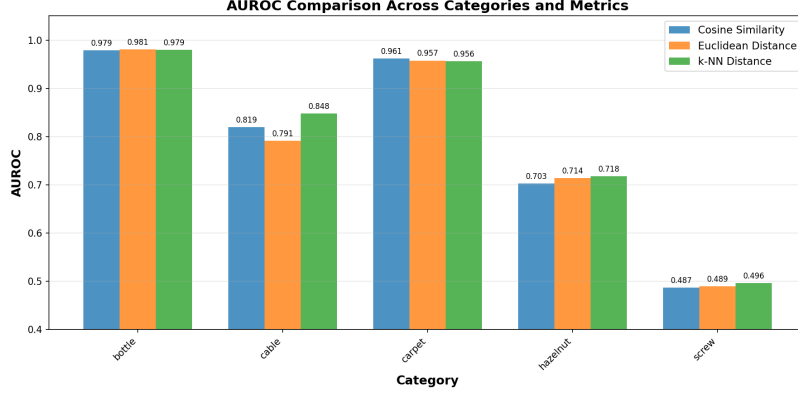


Fig. A.3: AUROC Score of the cosine, euclidian and k-nn similarity scores for all 5 categories.

D. CONFUSION MATRIX

In industrial quality control, the cost of a False Negative (sending a defective product to a customer) is typically much higher than a False Positive (rejecting a good product, which can be re-checked).

Category	Zero-Shot				Few-Shot				Mirroring				Prompt + 5-shot				Prompt (Full)			
	TP	FP	TN	FN	TP	FP	TN	FN	TP	FP	TN	FN	TP	FP	TN	FN	TP	FP	TN	FN
Bottle	30	0	20	33	56	0	20	7	56	0	20	7	56	0	20	7	62	1	19	1
Cable	33	6	52	59	63	10	48	29	68	18	40	24	63	10	48	29	87	10	18	2
Carpet	82	15	13	7	65	0	28	24	66	2	26	23	65	0	28	24	87	17	41	5
Hazelnut	64	32	8	6	54	16	24	16	57	26	14	13	54	16	24	16	66	10	30	4
Screw	54	14	27	65	84	23	18	35	22	0	41	97	84	23	18	35	114	27	14	5

Table A.1: Confusion Matrix values (TP, FP, TN, FN) for all methods across the five evaluated categories.

Analysis of Confusion Matrix Results: Table A.1 reveals a high rate of False Negatives (FN) in the Zero-Shot baseline, particularly for high-variance categories like 'Screw' (65 FN) and 'Cable' (59 FN). Few-Shot adaptation significantly mitigates this, reducing FNs by approximately 46% in the 'Screw' category. The Mirroring baseline proves highly effective for symmetric objects (e.g., 'Bottle') but fails on asymmetric objects, as shown by the high FN count (97) for 'Screw'.

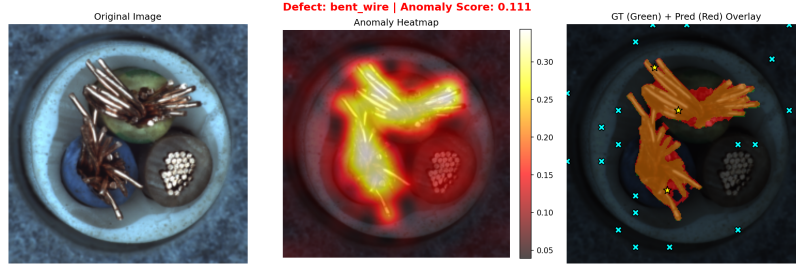
E. RECALL AND PRECISION

Category	Zero-Shot		Few-Shot		Mirroring		Prompt + 5 Shot		Prompt (Full)	
	Rec	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec	Prec
Bottle	0.48	1.00	0.88	1.00	0.89	1.00	0.89	1.00	0.98	0.98
Cable	0.36	0.85	0.68	0.86	0.74	0.79	0.68	0.86	0.95	0.83
Carpet	0.92	0.85	0.73	1.00	0.74	0.97	0.73	1.00	0.98	0.90
Hazelnut	0.67	0.65	0.77	0.77	0.81	0.69	0.77	0.77	0.94	0.86
Screw	0.45	0.79	0.71	0.79	0.18	1.00	0.71	0.79	0.96	0.81
Average	0.58	0.83	0.75	0.88	0.67	0.89	0.67	0.75	0.96	0.88

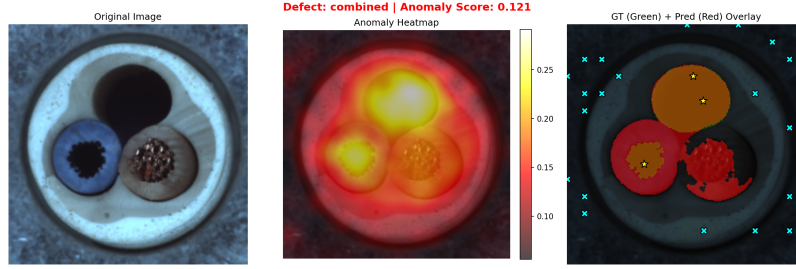
Table A.2: Comparison of Recall (Rec) and Precision (Prec) across different methods for MVTEC AD categories.

Analysis of Recall and Precision Results: For the Prompt-based methods (Table A.2), the 'Prompt (Full)' configuration yields the highest Average Recall (0.88), suggesting it is the optimal choice for minimizing defect escapes in an industrial setting. The 'Prompt + 5-shot' variation has a strong Average Recall (0.75), matching the performance of Few-Shot while requiring significantly less storage for reference embeddings.

F. SUPPLEMENTARY FIGURES

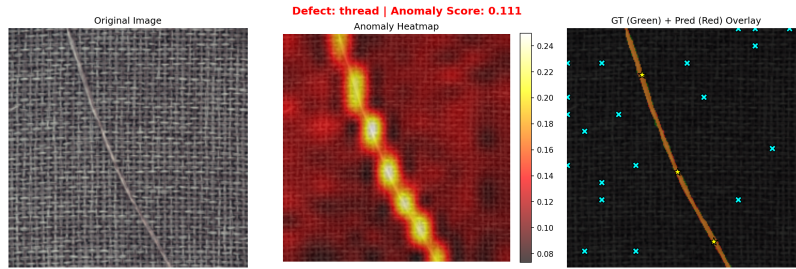


(a) Cable Successful Segmentation

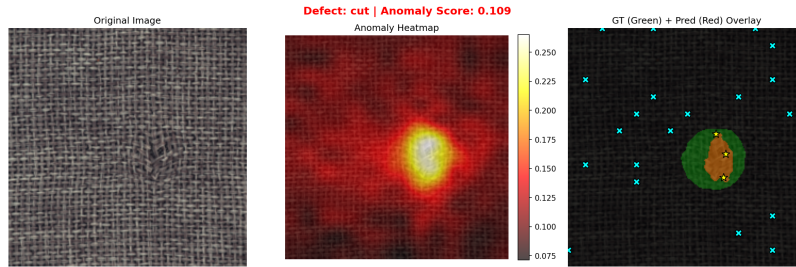


(b) Cable Poor Segmentation

Fig. A.4: Qualitative evaluation of SAM segmentation performance on the 'cable' category. **(a)** An example of accurate segmentation where SAM successfully masks the bent wires. **(b)** A poor performance case where the heatmaps are too diffused, causing SAM to over segment the cables rather than localizing the specific defects.

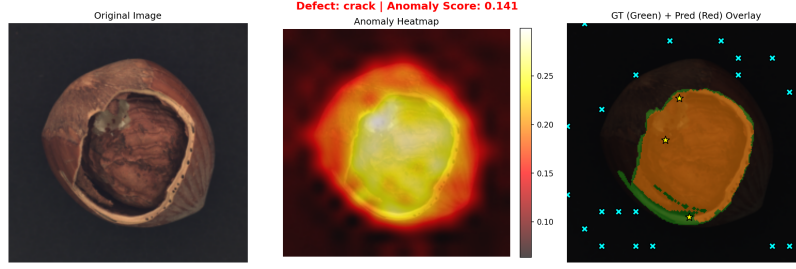


(a) Carpet Successful Segmentation

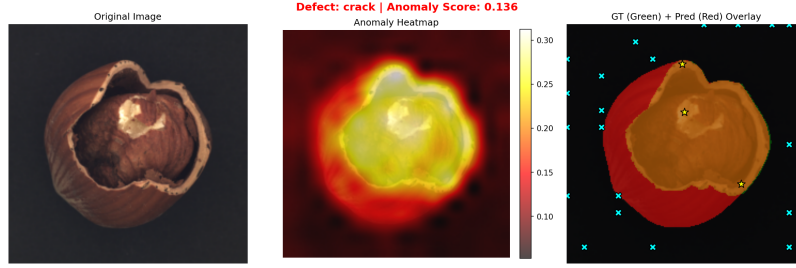


(b) Carpet Poor Segmentation

Fig. A.5: Qualitative evaluation of SAM segmentation performance on the 'carpet' category. **(a)** An example of accurate segmentation where SAM successfully masks the thin thread. **(b)** A poor performance case where the prompt points are too clustered, causing SAM to under segment damaged fabric.



(a) Hazelnut Successful Segmentation



(b) Hazelnut Poor Segmentation

Fig. A.6: Qualitative evaluation of SAM segmentation performance on the 'hazelnut' category. **(a)** An example of accurate segmentation where SAM successfully masks the cracked shell. **(b)** A poor performance case where SAM segments the entire hazelnut likely due to the unbroken shell being below the threshold for negative point prompts.

G. SIMILARITY METRICS

$$\text{CosineSimilarity}(x, y) = \frac{x \cdot y}{\|x\| \|y\|} \quad (1)$$

$$\text{L2Distance}(x, y) = \|x - y\|_2 = \sqrt{\sum_i (x_i - y_i)^2} \quad (2)$$

$$\text{kNNDistance}(x) = \frac{1}{k} \sum_{i=1}^k \|x - y_i\|_2 \quad (3)$$

H. DELIVERABLES

All the code is made publicly available at <https://github.com/Rian-Go/Hand-in-Repo-Deep-Learning>. Notebook to run models and reproduce results located at https://github.com/Rian-Go/Hand-in-Repo-Deep-Learning/blob/main/main_results.ipynb

DECLARATION OF USE OF GENERATIVE AI

This declaration **must** be filled out and included as the **final page** of the document. The questions apply to all parts of the work, including research, project writing, and coding.

- I/we have used generative AI tools: [yes / no]

If you answered yes, please complete the following sections. List the generative AI tools you have used:

- VS Copilot
- Chat GPT
- Claude Code

Describe how the tools were used:

What did you use the tool(s) for?

- Brainstorming
- Code Implementation
- Figure generation

At what stage(s) of the process did you use the tool(s)? Throughout all stages, but always with a human in the loop.

How did you use or incorporate the generated output? Used for structural generation of figures and result tables.