UniSpector: Towards Universal Open-set Defect Recognition via Spectral-Contrastive Visual Prompting

Appendix

1. Dataset Details

We construct the *Inspect Anything* (InsA) benchmark from seven industrial inspection datasets: GC10-DET [7], Magnetic Tile Surface Defect [3], Real-IAD [8], MVTec AD [2], 3CAD [11], VISION [1], and VisA [12]. Among these, GC10-DET, Magnetic Tile Surface Defect, Real-IAD, and MVTec AD are used as *in-domain* datasets, with defect categories partitioned into *seen* classes for training and *unseen* classes for testing. In contrast, 3CAD, VISION, and VisA serve as *cross-domain* datasets, used exclusively for evaluating transferability to novel defect types and object appearances. To ensure robust estimates of generalization performance, we form three independent seen–unseen splits with random seeds $\{42, 82, 777\}$, holding out roughly 25% of defect categories in each in-domain dataset as *unseen*.

1.1. Dataset Composition and Characteristics

Table 1 summarizes key characteristics of each dataset, including product types, materials, and HSV color statistics. The InsA benchmark is composed of diverse products and materials, enabling evaluation of open-set visual inspection performance across varied distributions. Defect images are captured under a mix of distinct imaging conditions (e.g., very bright or very dark), resulting in generally large standard deviations in HSV channels. These distributions also vary across datasets, leading to differing mean values and further highlighting the challenge of generalizing visual inspection models across heterogeneous industrial data.

GC10 [7]. GC10-DET is a defect detection dataset collected from metallic surfaces in industrial environments. It comprises roughly 2,300 defect images annotated with bounding boxes for ten defect categories: *silk spot*, *welding line*, *punching hole*, *water spot*, *crescent gap*, *oil spot*, *inclusion*, *waist folding*, *crease*, and *rolled pit*. For each random seed, we select three categories as *in-domain unseen* test set and use the remaining seven categories as *in-domain seen* training set under that seed.

MagneticTile [3]. The magnetic tile surface defect dataset contains images of magnetic tile surfaces collected from industrial production lines, covering both defective and defect-free samples. It provides pixel-level annotations for five defect categories—*blowhole*, *crack*, *break*, *fray*, and *uneven*—while normal images are discarded in our benchmark to focus solely on faulty cases. We retain approximately 400 defect images from the original release. For each random seed, we select one defect category as an *in-domain unseen* test set and use the remaining four categories as *in-domain seen* classes for training under that seed.

Real-IAD [8]. Real-IAD is a multi-view industrial anomaly dataset comprising *30* real-world objects fabricated from diverse materials (e.g., plastic, rubber). Each object is captured under five viewpoints and exhibits *2–5* distinct defect modes selected from eight defect families (i.e., *pit*, *deformation*, *abrasion*, *scratch*, *damage*, *missing parts*, *foreign objects*, and *contamination*). Only viewpoints equipped with pixel-level defect masks are retained as anomaly samples, and we use the official 1024×1024-resolution version of the images. From the binary defect masks, we perform connected-component labeling with 8-connectivity to obtain instance-level segments, and discard components whose width or height is smaller than 1% of the image width or height to remove tiny noisy polygons. For each split, we select *28* defect categories as *in-domain unseen* test classes and use the remaining categories as *in-domain seen* classes for training.

MVTec AD [2]. MVTec AD is a real-world industrial anomaly dataset that contains defective images across multiple object and texture categories, with various defect types such as contamination, oil, cuts, and cracks. In our benchmark, we discard all "good" images and use only anomalous samples with polygon masks. From these masks, we extract individual defect instances and remove components whose width or height is smaller than 1% of the corresponding image dimension, thereby filtering out tiny noisy regions.

3CAD [11]. 3CAD is a large-scale anomaly detection dataset collected from real 3C product manufacturing lines, covering representative defects that arise in practical production environments. It focuses on parts made of three common materials (Aluminum, Iron, and Copper) and includes multiple types of 3C components (e.g., camera cov-

Dataset	Product	Material	Color Distribution Statistics		
			H	S	V
GC10 [7]	Steel	Steel	0.0 ± 0.0	0.0 ± 0.0	85.7 ± 39.6
MagneticTile [3]	Magnetic Tile	Steel	0.0 ± 0.0	0.0 ± 0.0	109.4 ± 47.4
Real-IAD [8]	PCB, Toy Brick, Transistor, etc.	Plastic, Rubber, Wood, etc.	33.4 ± 49.5	40.6 ± 65.7	110.2 ± 105.5
MVTec AD [2]	Cable, Hazelnut, Tile, etc.	Glass, Metal, Fabric, etc.	46.7 ± 55.1	46.0 ± 47.8	120.5 ± 66.7
3CAD [11]	Camera Cover, Tablet PC, etc.	Aluminum, Copper, etc.	7.2 ± 19.2	16.7 ± 49.0	78.6 ± 70.5
VISION [1]	Capacitor, Lens, Screw, etc.	Plastic, Steel, Wood, etc.	25.2 ± 41.1	38.4 ± 70.7	106.3 ± 83.3
VisA [12]	Candle, Capsule, Macaroni, etc.	Plastic, Food, etc.	47.7 ± 36.4	95.7 ± 71.9	101.9 ± 66.1

Table 1. Dataset statistics used in the InsA benchmark, including their product types, materials, and the color distribution statistics (mean $\pm$ std) for each HSV channel.

ers, tablets, and PCs). Since 3CAD is designed for anomaly detection, we discard the "good" class without polygon annotations and also remove the *Multiple-defects* class, whose defect instances cannot be assigned to specific categories. After this filtering, we obtain a total of *46* defect categories. From the remaining polygons, we extract individual defect instances and discard components whose width or height is smaller than 1% of the corresponding image dimension to eliminate tiny noisy regions.

VISION [1]. The VISION benchmark unifies *14* industrial inspection subsets, each corresponding to a distinct object class from real manufacturing lines and captured at its native (often high) resolution; consequently, image size varies across subsets. The corpus provides pixel-level instance masks for *44* defect categories. For InsA, we retain only images in the *train* and *val* partitions that include polygon annotations and discard the *inference* split, whose labels are withheld. From the polygon masks, we remove segments whose width or height is smaller than 1% of the width or height of the image to filter out tiny noisy regions. All 44 defect categories are used exclusively for cross-domain evaluation, and none of them are included in the training data.

VisA [12]. VisA is an industrial visual inspection dataset containing normal and defective images from *12* object categories, some of which exhibit large variations in object location and pose across images. It covers both surface-level defects (e.g., scratches, dents) and structural defects (e.g., misplacement). In our benchmark, we discard all normal images and retain only those that contain at least one annotated defective region. We also exclude the *Other* defect class, whose semantics are not clearly specified. From the binary defect masks, we apply connected-component labeling with 8-connectivity to obtain instance-level segments, and discard components whose width or height is smaller than 1% of the corresponding image dimension, thereby filtering out tiny noisy regions.

1.2. Prompt Allocation

Selection of Prompt data To ensure class purity, prompt samples are selected only from *single-label* samples that contain the target defect alone; mixing co-occurring other categories into a prompt would inject ambiguous cues and dilute the class-specific signal.

Image-level vs. Instance-level Furthermore, unlike conventional few-shot protocols [5, 9, 10] that count shots per instance (box or mask), we define k with respect to entire *images*. In industrial defect inspection a single photograph often exhibits multiple defect instances whose visual similarity renders selective prompting ill-posed; excluding certain regions while prompting others would implicitly guide the detector to ignore genuine defects within the same frame. Counting shots at the image level therefore mirrors real deployment conditions and yields a fairer, more interpretable comparison across categories and methods.

Prompt Size For each defect category, we default to 5 visual prompt images during evaluation. However, to ensure a balanced evaluation across all dataset in InsA. For the "tail classes" (total images $\leq$ 10), we restrict the number of prompts to 1, as assigning 5 prompts to classes with very few target images leads to an irrational prompt-to-target ratio. As shown in Figure 1, by reducing the prompt to a single image for these cases, we achieved a more balanced and reasonable evaluation setting for long-tailed defect distributions. Notably, potential selection bias from using a single prompt is mitigated by averaging results across multiple random seeds, ensuring a robust evaluation.

2. Details of Radial Frequency Extraction

We present the radial frequency extraction procedure in Algorithm 1. Given a region crop, 2D FFT is computed and the resulting frequency magnitudes are aggregated into radial bins and L2-normalized. This process introduces minimal computational overhead while providing complementary texture cues to the spatial features.

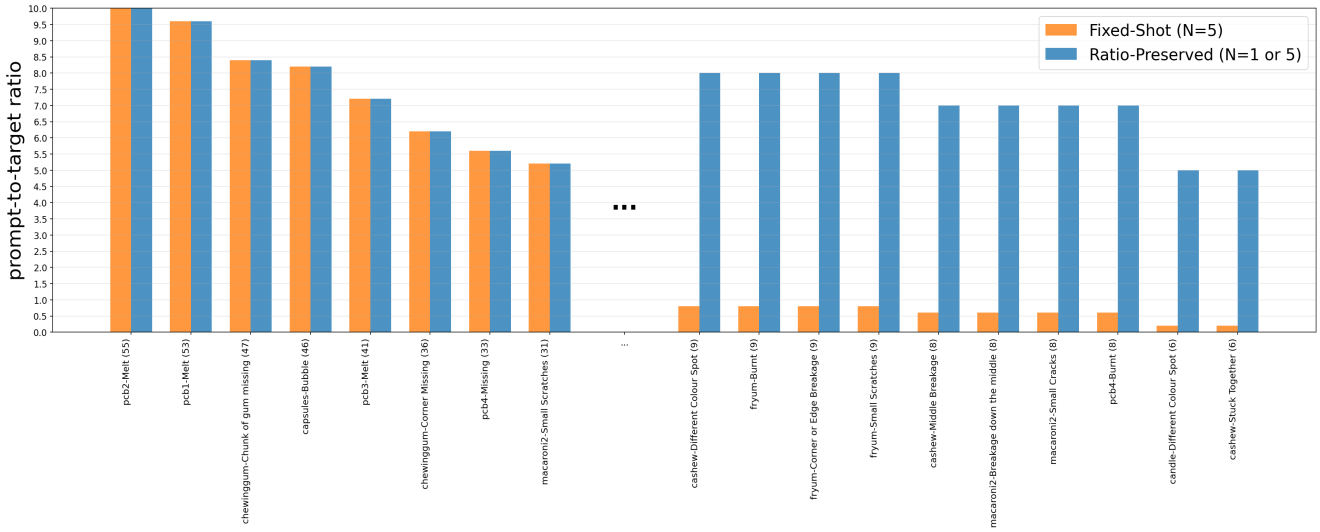

Figure 1. Distribution of Prompt-to-Target Ratios across Defect Classes. It illustrates the prompt-to-target ratio for each defect class (with sample sizes). We compare our adaptive prompt strategy (Blue) against the fixed prompt strategy (Orange). By reducing the number of prompts to one for tail classes (total images $\leq$ 10), the adaptive strategy prevents an irrational ratio in the tail classes, ensuring a more balanced and fair evaluation across the entire InsA dataset.

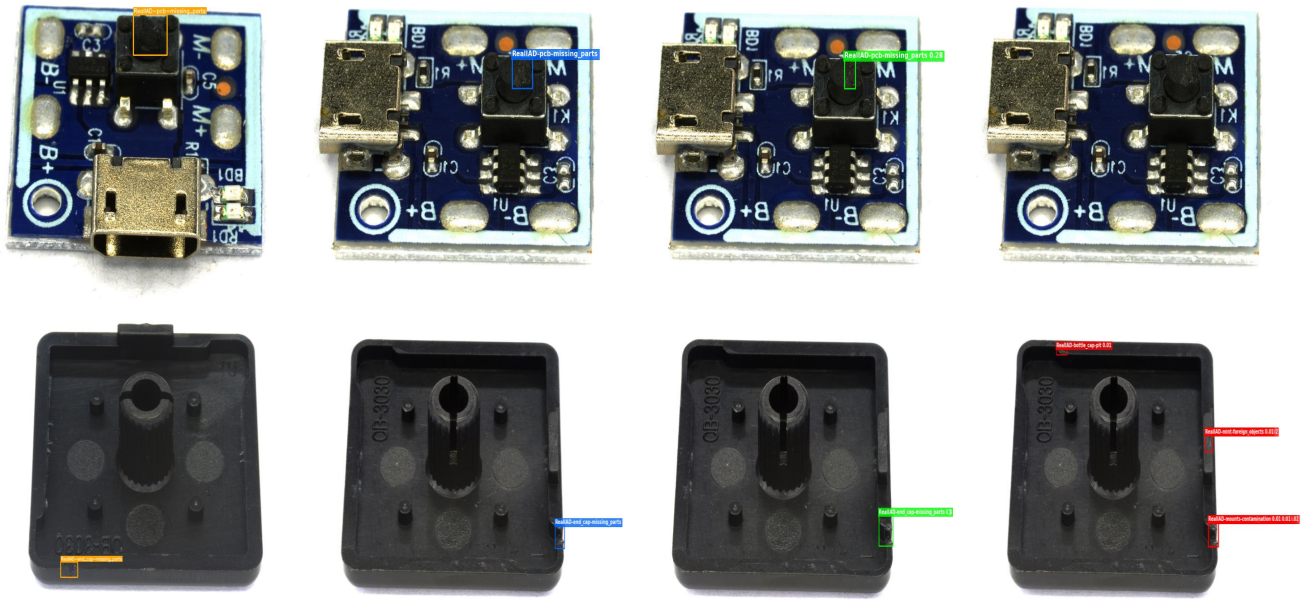

Figure 2. *UniSpector* is capable of recognizing unseen defects via visual prompts. (a) Orange box: user-specified prompt region. (b) Blue box: corresponding ground-truth in the target image. (c) Green boxes: correct predictions by *UniSpector*, accurately localizing subtle defects. (d) Red boxes: DINOv predictions, showing failure to localize the prompted defect.

Algorithm 1 Radial Frequency Feature Extraction

Require: Image I, Bounding boxes $\mathcal{B}$, Frequency bins J
Ensure: Spectral features $\mathbf{F} \in \mathbb{R}^{K \times J}$

for each bounding box $b_i \in \mathcal{B}$ **do**
  $I_{\text{crop}} \leftarrow \text{Crop}(I, b_i)$ and convert to grayscale
  $M \leftarrow |\text{FFTShift}(\text{FFT2D}(I_{\text{crop}}))|$
  $R \leftarrow \sqrt{(X - c_x)^2 + (Y - c_y)^2}$
  $R_{\text{norm}} \leftarrow R / \max(R)$
  for $j = 1$ to J **do**
    $f_j \leftarrow \text{mean}(M[R_{\text{norm}} \in \text{bin}_j])$
  end for
  $\mathbf{f}_i \leftarrow \text{L2Normalize}([f_1, ..., f_J])$
end for
return $\mathbf{F} = [\mathbf{f}_1, ..., \mathbf{f}_K]^T$

3. Implementation Details

3.1. Architectural Details

UniSpector is built upon the MaskDINO [6] architecture, an encoder-decoder framework designed for unified object detection and segmentation. For the encoder, we employ a Swin-T backbone with hierarchical stages to extract multi-scale feature maps. For fair comparison, the same SwinT backbone is used for DINOv, T-Rex2 and *UniSpector*, while YOLOE utilize its native YOLO11m [4] backbone due to architectural constraints. The decoder is composed of 9 Deformable Transformer layers with 8 attention heads. Each layer consists of self-attention, multi-scale deformable cross-attention, and an FFN.

3.2. Training Objective

We follow the standard MaskDINO training recipe, incorporating denoising (DN) training and auxiliary deep supervision across all L decoder layers to accelerate convergence. Bounding box regression is supervised by L_1 and GIoU losses, while mask segmentation employs a combination of BCE and Dice losses. Classification is addressed by a Focal loss measuring the similarity between object queries and class prototypes. The bipartite matching between predictions and ground truths is established via Hungarian matching. Finally, the proposed $\mathcal{L}_{\text{CPE}}$ is integrated with these standard terms to form total objective:

$$\mathcal{L}_{\text{total}} = \sum_{l=0}^{L-1} \Big(\lambda_{\text{cls}} \mathcal{L}_{\text{cls}}^{(l)} + \lambda_{\text{L1}} \mathcal{L}_{\text{L1}}^{(l)} + \lambda_{\text{giou}} \mathcal{L}_{\text{giou}}^{(l)} + \lambda_{\text{bce}} \mathcal{L}_{\text{bce}}^{(l)} + \lambda_{\text{dice}} \mathcal{L}_{\text{dice}}^{(l)} \Big) + \mathcal{L}_{\text{DN}} + \lambda_{\text{CPE}} \mathcal{L}_{\text{CPE}} \tag{1}$$

where each λ denotes the corresponding loss weight. Note that $\mathcal{L}_{\text{DN}}$ is the denoising counterpart of the same base losses with the same coefficients as their corresponding main losses. Under this joint optimization, we use the default weights: $\lambda_{\text{cls}} = 4$, $\lambda_{\text{L1}} = 5$, $\lambda_{\text{giou}} = 2$, $\lambda_{\text{bce}} = 5$, $\lambda_{\text{dice}} = 5$, and $\lambda_{\text{CPE}} = 1$.

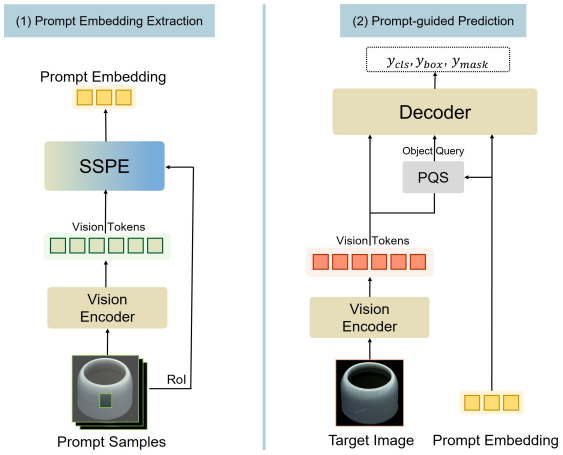

Figure 3. Detailed view of the inference phase. Unlike the training phase (see Fig. 3 in the main manuscript), inference is decoupled into "Prompt Embedding Extraction" and "Prompt-guided Prediction" to minimize computational overhead by leveraging cached embeddings.

3.3. Details of Inference Phase

As illustrated in Figure 3, *UniSpector* decouples the inference process into two distinct stages to minimize computational overhead. Specifically, Prompt Embedding Extraction is performed as a one-time offline process to extract prompt embeddings. During the subsequent Prompt-guided Prediction stage, these stored embeddings are loaded directly, bypassing redundant re-extraction of prompt embeddings for every query. Notably, this architecture allows *UniSpector* to seamlessly integrate prompt embeddings for any defect category of interest, enabling flexible and efficient deployment in open-set scenarios.

3.4. Training settings

All Swin-T-based models are initialized with the official DINOv pre-trained wights, while YOLOE is initialized from its publicly available checkpoint. Training is conducted for 20,000 steps with a total batch size of 48 on two NVIDIA H100 GPUs. We use the AdamW optimizer with a linear learning rate schedule, a base learning rate of 1×10^{-4}, a 10-step warm-up, and a weight decay of 0.05. Data augmentation includes random horizontal flipping, contrast adjustment (0.8–1.1), and brightness adjustment (0.5–1.3). All models are trained at an input resolution of 720×720 pixels. The radial frequency distribution is computed using 256 bins. The scaling factor is set to $s = 30.0$ and the angular margin to $m = 0.5$, which empirically stabilizes the optimization. We use $N_q = 300$ object queries and set the

Gumbel-Softmax temperature to $\tau = 1.0$.

4. Additional analysis

4.1. Robustness to the quality of prompts

To analyze robustness to prompt quality which is indeed critical factor for practical deployment. We conducted additional experiments simulating realistic industrial scenarios along two dimensions: image and annotation.

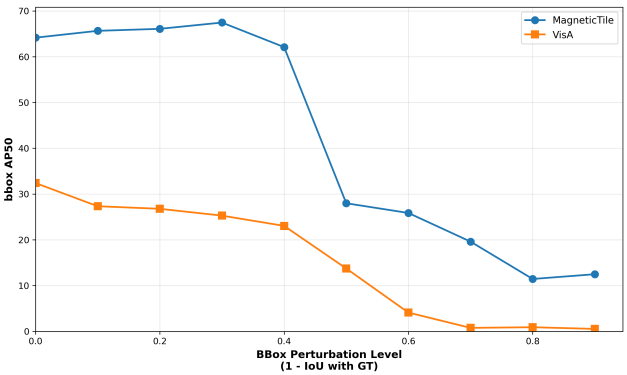

Figure 4. Robustness against prompt annotation (averaged over 10 trials per IoU). Each perturbed box is constrained to a specific IoU level relative to the ground-truth label.

Table 2. Experimental analysis under Gaussian Blur on the prompt images. We evaluate the impact of varying blur levels (σ) on the performance.

Dataset	Metric	Clean	0.5σ	1.0σ	1.5σ	2.0σ	2.5σ	3.0σ
VisA	AP^b_{50}	32.5	33.3	32.0	30.0	27.4	25.8	23.4
	AP^m_{50}	28.2	28.3	27.6	26.3	23.8	22.3	20.4
MTile	AP^b_{50}	64.2	64.9	64.1	63.5	62.0	60.1	57.0
	AP^m_{50}	38.9	39.1	38.8	37.8	36.4	34.3	31.0

Image Degradation To simulate blurry conditions, we evaluated the performance under Gaussian blur kernels with controlled standard deviations (σ) on the prompt images. The results are summarized in Table 2. Interestingly, a slight performance improvement was observed at 0.5σ blur. This improvement likely stems from the suppression of high-frequency noise, yielding prompt embeddings that are less biased toward instance-specific artifacts. The performance differences remain small up to 1.5σ, while a clear degradation is observed beyond 2.0σ, where the image quality degradation becomes visually noticeable. Performance is stable under Gaussian blur up to $\sigma = 1.5$ (Tab. 2) and box perturbations down to IoU 0.3 (Fig. 4, proving robustness.

Noisy Annotation We further examined robustness by randomly perturbing the prompt bounding boxes to achieve IoU levels ranging from 0.1 to 0.9 relative to the ground truth (GT). Each configuration was averaged over 10 independent trials. As shown in Figure 4, UniSpector maintains high performance down to an IoU of 0.3, demonstrating significant tolerance to imprecise annotations. Beyond this threshold, the perturbed boxes begin to encompass background or non-defect regions rather than the target defect, leading to an expected drop in precision as the prompt's semantic signal becomes corrupted.

4.2. Analysis of In-Context Learning Approaches

We conducted quantitative comparisons with representative in-context learning methods, PerSAM[1] and VRP-SAM[2], as shown in the table below. When evaluated in a training-free manner, PerSAM achieved an average AP^m_{50} of 6.0 due to the significant domain gap. The fine-tuning variant (PerSAM-F) yielded only a marginal improvement of +1.4 in average performance. We attribute this to a representation bottleneck: both PerSAM and VRP-SAM fundamentally rely on the representational capacity of frozen SAM encoders. Relying on frozen SAM general-object representations, PerSAM and VRP-SAM fail to generalize to InsA even after fine-tuning. Because the feature space of SAM, which is primarily trained on general real-world objects, may not be optimally suited for discriminating fine-grained features of subtle and complex defects. For VRP-SAM, since the authors do not release pre-trained weights for the VRP-Encoder, we report only the performance after training the VRP-Encoder on our in-domain seen set. Relying on frozen SAM general-object representations, PerSAM and VRP-SAM fail to generalize to InsA even after fine-tuning (Tab. 3).

Table 3. Results of in-context learning methods (AP^m_{50}) on InsA benchmark. † indicates results obtained by fine-tuning on the in-domain seen split of the proposed protocol.

Method	GC10	MTile	RIAD	MVTe	3CAD	VISN	VisA	Avg
PerSAM	0.0	33.0	0.8	4.8	0.1	0.3	3.0	6.0
PerSAM†	0.1	34.2	1.5	8.7	0.2	1.0	6.0	7.4
VRP-SAM†	0.4	12.5	22.3	18.8	3.2	4.7	12.2	10.6

4.3. Visualization

Figure 2 presents qualitative results for open-set defect detection. Given a user-specified region in the prompt images (orange bounding box), *UniSpector* successfully identifies corresponding defect instances in the target image. As illustrated in Figure 2(c), *UniSpector* accurately captures even subtle and fine-grained defects in PCB (TOP). Moreover, despite the pronounced rotational difference between the prompt and target defect instance, *UniSpector* consistently localizes the prompted defect type (BTM).

References

[1] Haoping Bai, Shancong Mou, Tatiana Likhomanenko, Ramazan Gokberk Cinbis, Oncel Tuzel, Ping Huang, Jiulong Shan, Jianjun Shi, and Meng Cao. Vision datasets: A benchmark for vision-based industrial inspection. *arXiv preprint arXiv:2306.07890*, 2023. 1, 2

[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*, pages 9592–9600, 2019. 1, 2

[3] Yibin Huang, Congying Qiu, and Kui Yuan. Surface defect saliency of magnetic tile. *The Visual Computer*, 36(1):85–96, 2020. 1, 2

[4] Glenn Jocher and Jing Qiu. Ultralytics yolo11, 2024. 4

[5] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8420–8429, 2019. 2

[6] Feng Li, Hao Zhang, Huaizhe Xu, Shilong Liu, Lei Zhang, Lionel M Ni, and Heung-Yeung Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3041–3050, 2023. 4

[7] Xiaoming Lv, Fajie Duan, Jia-jia Jiang, Xiao Fu, and Lin Gan. Deep metallic surface defect detection: The new benchmark and detection network. *Sensors*, 20(6):1562, 2020. 1, 2

[8] Chengjie Wang, Wenbing Zhu, Bin-Bin Gao, Zhenye Gan, Jiangning Zhang, Zhihao Gu, Shuguang Qian, Mingang Chen, and Lizhuang Ma. Real-iad: A real-world multi-view dataset for benchmarking versatile industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22883–22892, 2024. 1, 2

[9] Xin Wang, Thomas E Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. Frustratingly simple few-shot object detection. *arXiv preprint arXiv:2003.06957*, 2020. 2

[10] Xiaopeng Yan, Ziliang Chen, Anni Xu, Xiaoxi Wang, Xiaodan Liang, and Liang Lin. Meta r-cnn: Towards general solver for instance-level low-shot learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9577–9586, 2019. 2

[11] Enquan Yang, Peng Xing, Hanyang Sun, Wenbo Guo, Yuanwei Ma, Zechao Li, and Dan Zeng. 3cad: A large-scale real-world 3c product dataset for unsupervised anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9175–9183, 2025. 1, 2

[12] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. *arXiv preprint arXiv:2207.14315*, 2022. 1, 2