Kaputt: A Large-Scale Dataset for Visual Defect Detection

Sebastian Höfer[1] Dorian F. Henning[1] Artemij Amiranashvili[1] Douglas Morrison[1]
Mariliza Tzes[1] Ingmar Posner[1,2] Marc Matvienko[1] Alessandro Rennola[1]
Anton Milan[1]

[1]Amazon, Fulfillment Technologies & Robotics [2]University of Oxford, Applied AI Lab
{hoefersh,doriahen,artemija,morridou,mtzes,ingmarp,mrcmtv,arenno,antmila}@amazon.com

Abstract

We present a novel large-scale dataset for defect detection in a logistics setting. Recent work on industrial anomaly detection has primarily focused on manufacturing scenarios with highly controlled poses and a limited number of object categories. Existing benchmarks like MVTec-AD [6] and VisA [33] have reached saturation, with state-of-the-art methods achieving up to 99.9% AUROC scores. In contrast to manufacturing, anomaly detection in retail logistics faces new challenges, particularly in the diversity and variability of object pose and appearance. Leading anomaly detection methods fall short when applied to this new setting. To bridge this gap, we introduce a new benchmark that overcomes the current limitations of existing datasets. With over 230,000 images (and more than 29,000 defective instances), it is 40 times larger than MVTec and contains more than 48,000 distinct objects. To validate the difficulty of the problem, we conduct an extensive evaluation of multiple state-of-the-art anomaly detection methods, demonstrating that they do not surpass 56.96% AUROC on our dataset. Further qualitative analysis confirms that existing methods struggle to leverage normal samples under heavy pose and appearance variation. With our large-scale dataset, we set a new benchmark and encourage future research towards solving this challenging problem in retail logistics anomaly detection. The dataset is available for download under https://www.kaputt-dataset.com.

1. Introduction

Automated visual defect detection is critical for quality assurance in numerous industrial and logistics processes. Particularly at the scale of large retailers that handle millions of unique items, accurate detection of anomalies can significantly reduce costs, minimize waste, and enhance overall operational efficiency. However, developing robust visual defect detection systems in retail logistics applications presents significant challenges that have yet to be fully addressed by existing research. The primary challenge stems from the diversity of items and the rarity of defects, which makes building supervised-learning datasets costly and time-consuming. This scarcity of training data leads to highly imbalanced datasets, necessitating unsupervised and anomaly-detection (AD) approaches.

State-of-the-art unsupervised and AD methods for visual defect detection achieve exceptional performance under controlled manufacturing conditions, reaching 99.9% [7] and 99.5% [30] AUROC on MVTec-AD [6] and VisA [33] datasets, respectively. However, these methods struggle in complex logistics environments like Amazon's retail operations, where millions of diverse products flow through logistics centers. The challenges are multifaceted (Figure 1): products range from consumables to electronics, each with distinct physical properties; defects vary from minor creases to major spillages, often with subtle manifestations that challenge even human inspectors; most items are observed only a few times, limiting both defective and non-defective sample availability; and significant pose variation occurs due to random product placement.

To enable researchers to overcome these challenges, we introduce a novel large-scale dataset for *visual defect detection in retail logistics applications*. Our dataset significantly advances the field by addressing key limitations of existing benchmarks and poses the following key question: How can we build generalizable visual defect detection methods under challenging conditions such as limited instances per item, limited availability of both defective and non-defective samples per item, and significant intra-class variation?

Our key contribution is a challenging defect detection dataset with unparalleled scale and diversity of products, structured to enable the development of novel supervised, unsupervised, and hybrid approaches. The dataset comprises 238,421 images of 48,376 unique items. Items are

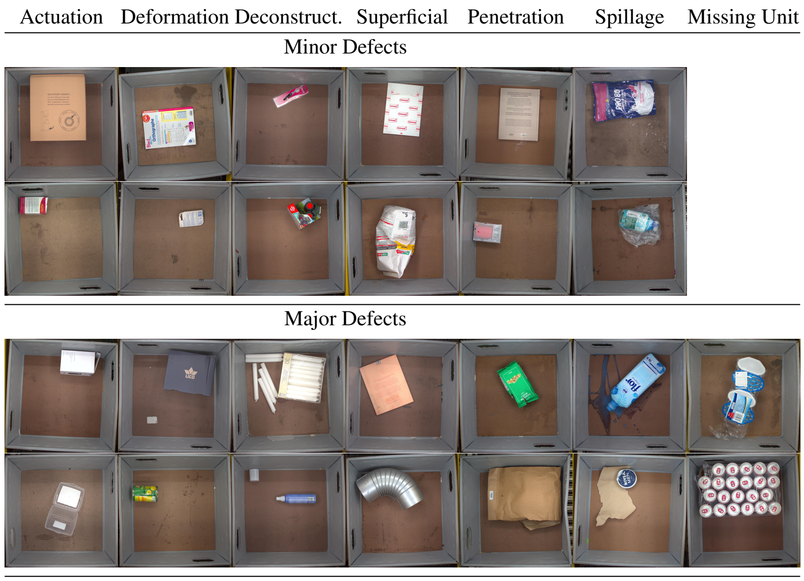

Figure 1. Overview of defect severities and defect types. Our dataset categorizes defective samples into two *severity* classes: *minor* (top two rows) and *major* (bottom two rows). Additionally, each defective sample is assigned one or multiple defect *types* (columns), which characterize the defect(s) an item exhibits in a more fine-grained manner. The figure shows two representative samples per defect type/severity combination.

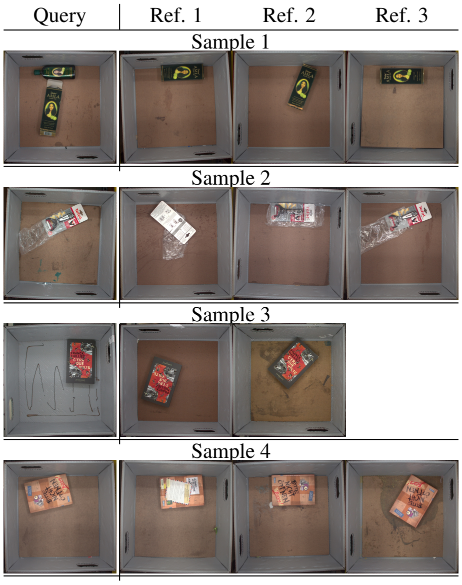

Figure 2. Each *query* image is associated with 1-3 *reference* images which may exhibit significant variability: (1) Benign case. (2) Defective reference image (< 1% of all reference images). (3) Significant background variation, and < 3 reference images available. (4) Pose variability (front vs. back).

presented in random poses and orientations, closely mirroring real-world retail logistics scenarios. The dataset is split into annotated and unannotated portions: the annotated *query image* dataset contains 100,267 images, including 29,316 defective instances. For all query images, we provide qualitative defect severity and fine-grained defect type annotations, reflecting the subjectivity present in defect assessment. Additionally, each query image is associated with up to three unannotated *reference images* that depict items in a “normal“ condition (Figure 2). We feature diverse product categories, seven distinct defect types, and high-resolution images capturing obvious and subtle defects.

To substantiate the challenge posed by our dataset, we evaluate numerous state-of-the-art baselines. First, we demonstrate that supervised baselines [10, 12] with access to a large training set of defective instances achieve up to 94.27% AUROC. These methods achieve high performance by learning common defect pattern priors, while struggling in edge cases and “adversarial” items, such as items with damage-like designs (*e.g.* a printed hole) or deformable items where creases may arise naturally without negatively impacting the product. We then demonstrate that these supervised methods fall short of yielding such performance under more realistic conditions, when only few defective samples are available for training. In such scenarios, unsupervised and anomaly detection baselines [13, 23] were shown to excel [30] on related datasets like MVTec-AD [6] or VisA [33]. However, we demonstrate that these methods, as well as state-of-the-art vision-language models [1, 4], fail to surpass 56.96% AUROC on our dataset. Qualitative analyses confirm that these methods struggle with item and pose variability as well as limited access to non-defective samples of the query item.

These results underscore the relevance of our dataset to the anomaly detection community in developing more robust and generalizable methods. By introducing this comprehensive dataset, we aim to stimulate progress in visual defect detection for retail logistics applications. We believe this unique resource will enable researchers and practitioners to develop more robust and generalizable models, capable of handling the complexities and nuances of real-world defect detection tasks. The dataset is available for download under https://www.kaputt-dataset.com.

2. Related Work

Defect detection applications. Defect detection is an important and widely studied field due to its many commercial applications, including detecting defective parts in industrial manufacturing [6], inspecting civil infrastructure such as bridges [24, 27], vehicle damage [32], and medical applications [15]. However, our use case differs from the standard industrial manufacturing applications, mainly in terms

Table 1. Overview of representative defect and anomaly detection datasets. Our dataset provides a unique new challenge to the defect detection field due to the amount of defective samples and intra-class variance within the dataset.

Dataset	**Labeled Samples**: Total (Anomalous)	**Item Categories**	**Unlabeled Samples**	**Defect Labels**	**Pose/Viewpoint Variance**
ARMBench [18]	100,000+ (6,786)	N/A	-	Classes	**yes**
Kolektor [28]	399 (52)	-	-	Classes	no
BTAD [17]	2830 (1799)	3	-	Classes	no
MVTec-AD [6]	5,354 (1,258)	15	-	Classes	no
VisA [33]	10,821 (1,200)	12	-	Classes, Segmentations	no
Ours	100,267 **(29,316)**	**48,376**	138,154	Classes	**yes**

of item variation and defect variability. While industrial applications typically focus on a single, known item or part, we are concerned with the much more open-ended problem of detecting defects for the millions of constantly changing items handled in retailers like Amazon, which may also exhibit significant intra-class variation (*e.g.* packaging variations and random poses). Thus, our work differs from much of the literature, in terms of data requirements and methods.

Datasets. The variety of defect detection applications has led to the development of a number of bespoke datasets in this domain [3, 11]. Most relevant to our application is ARMBench [18]. While targeting a similar domain and comparable in total size, ARMBench only contains one quarter of the defective samples our dataset offers, and features only two (open and deconstruction) compared to seven defect types. Strongly related are datasets targeting manufacturing defects, such as MVTec-AD[6] and VisA [33]. These contain images of items with a wide variety of defects such as dents, contaminations, and structural changes. At this point however, the performance on these datasets is close to being saturated, with state-of-the-art methods achieving well over 99% AUROC [30]. Our dataset offers one order of magnitude more data both in terms of annotations and anomalous instances, enabling researchers to leverage the dataset for developing and benchmarking various types of approaches. At the same time, significant pose variation render the dataset significantly more challenging than related ones. Table 1 presents a comprehensive comparison to existing defect detection datasets.

Models. The defect detection problem has been approached in different ways, using *supervised*, *unsupervised*, and *anomaly-detection* methods. The most straightforward approach is supervised learning, which aims to learn distinctive defect patterns given samples of both non-defective *and* defective instances. For these approaches, the supervision signal takes the form of an image label [24, 27, 32], a segmentation mask, or both [15], casting the machine learning problem as classification, segmentation or multi-task learning, respectively. A key limitation of these approaches is that they require access to a large dataset of *defective* items for training, which are typically rare and difficult to collect.

This limitation motivates the use of alternative approaches that can leverage non-defective, “normal” samples more effectively by identifying defective instances as deviations from the expected normal appearance. While the exact distinction between the underlying paradigms is blurry, these approaches are usually categorized as *unsupervised* learning, *anomaly* detection, or *outlier* detection. This includes methods that classify outliers directly given some representational space (*e.g.* using one-class SVMs [25, 31]) or those that threshold a per-pixel or per-image patch reconstruction error [5]. Similarly, deep generative approaches have been used to compute outlier statistics both on image reconstructions as well as in the learnt latent representation via Generative Adversarial Networks [2, 26], diffusion models [30], or pre-trained Vision Transformers [13]. Exemplar-based methods compute outliers directly by constructing a more targeted reference dataset on the fly, via a nearest-neighbour approach [23] and then computing image/patch-level feature distances relative to this set.

Such approaches are well suited to industrial applications, where examples of anomaly-free items in an identical, nominal pose are plentiful. However, they are prone to false positives predictions by flagging non-defect-related image variations as anomalous. As our experiments demonstrate, this makes current approaches impractical for real-world retail logistics applications where we are faced with significant intra-class variation, *e.g.* due to differing poses or packaging. More recently, Jiang et al. [14] investigated whether this limitation could be addressed by leveraging the inherent visual understanding capabilities of Multimodal Large Language Models (MLLMs). Their findings, however, demonstrate that current MLLMs’ performance falls short of industrial requirements: while excelling at *object* analysis and description tasks, these models lack robust *anomaly* detection capabilities. Our experiments using both commercial and open-source MLLMs [1, 4] corroborate these findings.

3. Dataset

Our dataset consists of top-down RGB images of retail items, each accompanied by categorical labels and segmentation masks. In the following sections, we detail the dataset's structure, collection, and annotation methodology.

3.1. Dataset Structure

The dataset is organized into *query* and *reference* sets:

1. *Query dataset*: Contains image captures with associated:
   (a) *Item identifier* (unique per item)
   (b) *Defect severity* (no defect, minor, major)
   (c) *Defect type(s)* for defective items (*e.g.*, penetration, spillage)
   (d) *Item material* (*e.g.* cardboard, plastic, books)
   (e) *Item segmentation mask*
2. *Reference dataset*: Contains 1-3 image captures per item identifier, primarily non-defective but not guaranteed.
   (a) *Item identifier* (unique per item)
   (b) *Item segmentation mask*

The dataset is further divided into training, validation, and test splits, each consisting of a unique query/reference set pair. To test model generalization capabilities and prevent overfitting to specific items, we ensure that each item only appears exclusively in one of the splits, i.e. identifiers do not overlap between splits.

3.2. Images

For image capture, we use a data collection station equipped with a 12 MP RGB camera with an f/12mm lens. The camera is positioned top-down to capture the singulated item located inside a logistics container ("tray"). To provide uniform diffuse illumination while minimizing reflections commonly induced by plastic materials, we enclose the station with side walls and ensure constant lighting using LED panels. We provide a schematic drawing of the data collection setup in the Supplementary Material (Section III).

We further post-process the acquired images by applying a square crop that includes only the tray, and resize the images to 2048 × 2048px. We also provide item segmentation masks/crops (Section 3.4.1), but retain the full tray images as item boundaries are not always clearly defined due to dangling or protruding parts, and certain defects may only be visible on the tray surface (*e.g.*, liquid spillages; see Figure 1, examples 1 and 3, respectively).

3.3. Data Collection

A major challenge in creating defect detection datasets is the rarity of defect events, making the acquisition of positive (defective) samples extremely time-consuming. We address this through a two-stage collection strategy: First, we collect items flagged as defective by human operators for annotation. Second, we implement an iterative mining process where a binary classifier, trained on previously annotated images (Section 3.4), identifies potential defect candidates for further annotation.

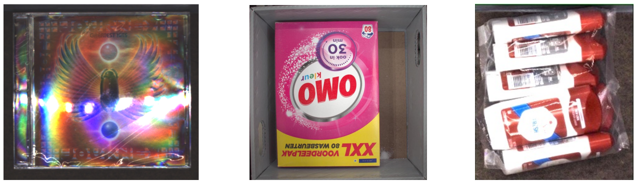

Figure 3. Examples for challenging defective cases (from left to right). (1) Unobservable cases. A small stripe in the bottom half of the CD could be either a reflection or a crack in the cover. (2) Complex cases. The detergent pack looks intact, but at a second look the powder on the tray next to it item indicates a spillage defect. (3) Ambiguous cases. The multi-pack is complete but its units are unordered, which is acceptable but has different visual appearance than the corresponding reference image.

The resulting initial query dataset of defective and non-defective images undergoes further curation based on the following criteria: (1) *Quality control* through manual filtering of low-quality images, particularly those with missing or off-center items. (2) *Diverse item range* with maximum 15 samples per item to ensure variety. (3) *Balanced defect rate* (28.6%) that aligns with existing benchmarks like MVTec-AD [6] and VisA [33] since defective samples are more valuable for training and evaluation than non-defective samples. (4) *Exclusion* of items lacking non-defective samples to prevent model overfitting.

The curated query dataset is split by item identifier into training (85%), test (10%), and validation (5%) sets, each supplemented with up to three reference images from a separate unlabeled dataset. We remove missing or off-center reference images but exclude defect type and severity labels. This approach, including the limit of three reference images per sample, reflects real-world retail conditions where most items sell infrequently and creating a perfect reference database is impractical. Consequently, some limited amount of reference images may exhibit different packaging or contain defects.

3.4. Annotation

Next, we describe the labels and annotation process.

3.4.1. Item Segmentation Masks

The images depict the item inside the full tray, but some baseline methods may require item crops to perform optimally. We thus generate item segmentation masks using a U-Net [22] model trained on 17,000 manually annotated masks, and create *square* item-crops with 10% padding. The generated masks and item-crop images are released as part of the dataset. Moreover, we evaluate the baselines on both full and item-cropped images.

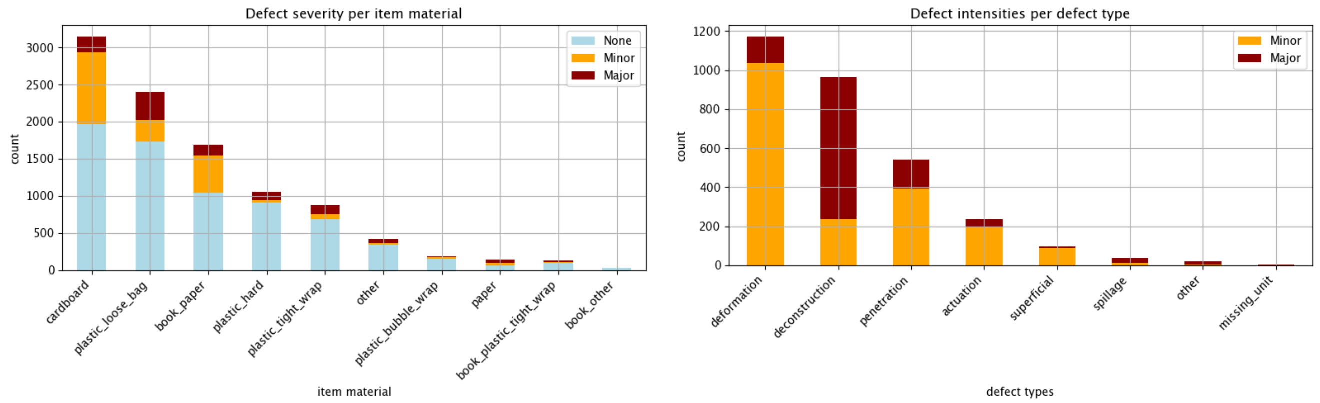

Figure 4. **Left**: Distribution of item material types and defect severities. We observe that items with cardboard material dominate the dataset, followed by plastic bags/cases and books. **Right**: Distribution of defect types per defect severity. We find that *deformation* is the most common defect type, however, it mostly results in minor defect severity, similar to *penetration*, *actuation* and *superficial*. In contrast, *deconstruction* and *spillage* commonly result in major defect severity.

3.4.2. Categorical Labels

Both the query and reference datasets are curated to avoid low-quality images, as described above. Additionally, each sample in the query datasets is manually annotated with the following categorical labels: *defect severity*, *defect type*, and *item material*.

Defect severity. Each sample is annotated with a defect severity label: *no defect*, *minor*, or *major*. Major defect compromises the item's integrity (*e.g*. significant crush or puncture) or risks doing so (*e.g*. fully opened box lid). Minor defect renders the item not pristine but potentially acceptable (*e.g*. small dents on cardboard packaging). Acknowledging the subtle boundaries between these categories, our benchmark (Sec. 4) focuses on detecting *any* defect (minor and major), with ablation studies model performance on major defect detection.

Defect type. For each defective sample exhibiting at least minor defect severity, we annotate one or more *defect types*: *penetration* (*e.g*. holes, tears, cuts), *deformation* (*e.g*. dents, crushes), *actuation* (*e.g*. open box/bag/book), *deconstruction*, *spillage* (liquid, powder, *etc*.), *superficial* (*e.g*. dirt, scratches), *missing unit*. Assigning multiple defect types per item is explicitly permitted, as items may incur multiple defects at the same or different spatial locations.

Item material. Each item is categorized according to its primary outer material: *cardboard*, *plastic (loose bag)*, *plastic (hard)*, *plastic (bubble wrap)*, *plastic (tight wrap)*, *paper*, *book (paper)*, *book (plastic)*, *other*. The distribution per item material is shown in Figure 4 (left), indicating higher volumes of cardboard and plastic packaging, followed by books.

Each sample is independently labeled by three annotators. We then aggregate annotations through majority voting. During our baseline evaluation experiments, we found some wrongly labeled samples, which we manually corrected. We observe that annotation errors primarily arise from the following issues, exemplified in Figure 3: (1) *unobservable* cases where defects cannot be detected due to sensing limitations or lack of non-defective reference images; (2) *complex* cases where defects are present but so subtle that they get overlooked by annotators; (3) *ambiguous* cases where an anomaly is visible but it is not clear whether it qualifies as a defect. We acknowledge that, despite best efforts, the dataset may still contain mislabeled samples, but demonstrate that these do not negatively affect the evaluated baselines (see Supplementary Material).

We visualize the resulting distribution of defect severities and types in Figure 4 (right)[1].

4. Benchmark

To demonstrate the challenge posed by our dataset and establish baseline performance, we evaluate various state-of-the-art models. We define four distinct evaluation scenarios, based on whether an approach uses the training data, the reference images, none or both. We choose methods in such a way to cover a wide variety of relevant approaches, favoring established and widely adopted methods over their latest variants.

4.1. Evaluation Metrics

To compare the different baselines, we formulate the task as a binary classification problem between *no defect* and

[1] The figure presents a slightly simplified version of the defect type distribution, as we only assume one single defect type per sample, which holds true for 72% of all defective samples in the dataset. For samples with multiple defect types, we select one based on a predefined priority list, where more severe defects (such as spillage) take precedence over less severe ones (such as actuation).

any defect (*i.e.* defect severity minor or major). Formally, we evaluate a classifier $f(\mathbf{x}_{\text{q}}^{\text{ID}}, \{\mathbf{x}_{\text{ref}}^{\text{ID}(1)}, \ldots, \mathbf{x}_{\text{ref}}^{\text{ID}(G_{\text{ID}})}\}) = \hat{y}$ given a (labeled) test query image $(\mathbf{x}_{\text{q}}^{\text{ID}}, y) \in \mathcal{D}_{\text{q}}^{\text{test}}$ with item identifier ID and binary defect label $y \in \{0, 1\}$ (0 = non-defective, 1 = defective), and a set of G_{ID} corresponding (unlabeled) test reference images with the same item identifier from $\mathcal{D}_{\text{ref}}^{\text{test}}$. Note that both the number of query and reference images per item identifier varies, and also that models (*e.g.* the methods listed in Section 4.2.3) may ignore reference images altogether.

Apart from the query and reference test sets, our dataset also comprises equivalent subset pairs for model training $\mathcal{D}_{\text{q}}^{\text{train}}$, $\mathcal{D}_{\text{ref}}^{\text{train}}$ and validation $\mathcal{D}_{\text{q}}^{\text{valid}}$, $\mathcal{D}_{\text{ref}}^{\text{valid}}$. The training datasets are used by the supervised and combined methods, while the validation datasets are used for hyperparameter tuning and decision threshold selection.

To compare different models f, we use Average Precision (AP) on **any** (minor or major) defect (**AP$_{\text{any}}$**), as our key performance metric, and additional auxiliary metrics:

- Average Precision (AP) on major defect (**AP$_{\text{major}}$**), computed only on the subset of either non-defective or items with major defects,
- Area under Receiver Operator Characteristic (**AUROC**),
- Recall at 50% Precision (**R@50%P**), and
- Recall at 1% False Positive Rate (**R@1%FPR**).

Some methods perform better on full images and others on item-cropped images. For the sake of brevity, we report numbers only for the best-performing variant of each method, indicating what type of image is used in Table 2.

4.2. Scenarios

We present four distinct evaluation scenarios that each explore unique aspects of our dataset.

(1) *No training and no reference images*. Such approaches are commonly referred to as zero shot, leveraging strong general-purpose vision-language models, here CLIP [20], Claude [4], and Pixtral [1].

(2) *No training and with reference images*. Here, models have no access to the labeled training set but they leverage (unlabeled) reference images at test time. In the anomaly detection (AD) context, this approach is commonly referred to as few-shot AD [31] or few-normal-shot AD [13].

(3) *With training and without reference images*. These are purely supervised methods [10, 12] that leverage the annotated training data to train a binary classifier, ignoring the reference datasets.

(4) *With training and with reference images*. These methods combine approaches (2) and (3) by both training a model (backbone) and leveraging reference images.

Next, we detail the methods we evaluate as part of each of the four scenarios. All model and training details required to replicate the results are provided in the Supplementary Material V. Note that some methods can be applied in multiple scenarios, as we point out in the following.

4.2.1. No Training and No Reference Images

In this scenario, we test whether and to which extent strong general-purpose image understanding capabilities translate to zero-shot defect detection performance.

CLIP. We test the vanilla CLIP model [20] (`CLIP`), and CLIP with fine-tuned prompts [21] (`POMP`). For vanilla CLIP, we perform manual prompt optimization on the validation set, and ended up with the following prompts for classification: `Image of an item without problems` and `Image of an item with problems`, for non-defective and defective samples, respectively. For POMP, we use the labels `undamaged` and `damaged` for the respective classes.

WinCLIP. WinCLIP [13] extends the original CLIP model for anomaly detection, by (1) providing a diverse set of text prompts representing defective and healthy samples, and (2) using multi-scale image feature extraction and comparison. In the zero-shot setting, WinCLIP only uses query image and text prompts (`WinCLIP-zero`).

Claude. We evaluate Anthropic's Claude 3.5 Sonnet [4], a public Vision-Language Model (VLM), in the zero-shot setting in two ways. `Claude-zero` evaluates the model when provided with a text prompt and the query image, and ask the model to inspect the image for defects using Chain-of-Thought and finally grade the defect severity on a scale from 0 to 10. We tested different prompts on a small held-out set, and apply the best-performing prompt to the entire dataset (see Supplementary Material I). In the second setting (`Claude-icl`), we apply the few-shot in-context learning (ICL) scenario, where we additionally provide five samples as positive *defective* classes with respective example answers. Due to computational constraints, prompt images are rescaled to 512×512.

Pixtral. In addition to Claude, we evaluate the recent open-source Pixtral-12B model [1] as another VLM on our dataset. Similarly, we evaluate both a zero-shot (`Pixtral-zero`), and an in-context learning (`Pixtral-icl`) setting. The best-performing prompts can be found in the Supplementary Material I.

4.2.2. No Training and With Reference Images

We evaluate two AD approaches that leverage reference images of known item categories at test time.

PatchCore [23] is a state-of-the-art anomaly detection method that leverages patch-level features from an image, comparing each patch's feature to a memory bank of normal patches and identifying anomalous samples through patch-level feature distance. The image-level anomalous score is computed as the maximum of the patch-level anomaly scores across all patches in the image. We use the reference

Baseline	Tray/Item	**AP**$_{\textbf{any}}$ [%]	**AP**$_{\textbf{major}}$ [%]	**AUROC**	**R@50%P** [%]	**R@1%FPR** [%]
`Random`	-	31.84	14.00	50.00	0.00	1.08
No training, no references (zero-shot, few-shot)						
`CLIP`	item	36.20	17.15	56.05	**0.56**	**1.53**
`POMP`	item	32.98	18.17	50.44	0.00	1.28
`WinCLIP-zero`	item	33.87	19.11	52.30	0.03	1.37
`Claude-icl`	tray	**36.57**	**24.76**	**56.96**	0.00	0.31
`Pixtral-zero`	tray	32.75	16.42	50.93	0.00	0.81
`Pixtral-icl`	tray	32.18	15.83	50.86	0.00	0.69
No training, *with* references (few-shot, non-parametric, in-context learning)						
`PatchCore50`	item	**35.86**	17.80	**54.69**	**2.46**	**2.18**
`WinCLIP-few`	item	34.05	**19.29**	52.41	0.66	1.56
***With* training, no references** (supervised/instruction fine-tuning)						
`ResNet50`	tray	81.06	74.93	88.36	91.98	30.01
`ViT-S`	tray	**90.67**	**91.45**	**94.27**	**97.69**	**59.36**
`Pixtral-ft`	tray	33.43	17.19	51.44	3.62	3.62
`AutoGluonMM`	item	87.77	86.10	92.47	96.76	46.26
***With* training, *with* references** (supervised with references, non-parametric with fine-tuning)						
`PatchCore50-ft`	item	40.18	20.98	60.14	6.52	2.37
`AutoGluonMM-ref`	item	**71.21**	**61.45**	**84.29**	**89.83**	**13.32**

Table 2. Results of the evaluated baseline methods on the test set split with 10067 total samples (minor defect: 2089, major defect: 1117).

test set to create the individual memory banks of features for different items and to compute the anomaly score. As a backbone for feature extraction, we test ResNet50 pretrained on ImageNet (`PatchCore50`).

WinCLIP. We test WinCLIP in the few-shot setting, by enabling it to perform visual feature comparison with reference images (`WinCLIP-few`).

4.2.3. With Training and No Reference Images

Here, we focus on common supervised methods that leverage the annotated training data to learn how to recognize the appearance of visual defects. To do so, we fine-tune two common types image backbones for defect classification using a binary cross entropy (BCE) loss, feeding only the query images as input.

Convolutional Networks. We fine-tune a ResNet50 model [12] backbone, pretrained on ImageNet [8], on our training data (`ResNet50`). Preliminary experiments demonstrated that a high resolution is necessary to prevent subtle or small defects from being obscured or lost due to downsampling, and we thus use a resolution of 1024×1024 pixels. Training is conducted for 20 epochs with an initial learning rate of 5×10^{-5} and batch size 48.

Vision Transformers. We fine-tune a ViT-small pretrained on DINOv2 [19] with patch size 14×14 px [9] at 1024×1024 px for 30 epochs, with an initial learning rate of 5×10^{-6} and batch size 8 (`ViT-S`). Additionally, we test an AutoML approach using the AutoGluon MultiModal framework [29] (`AutoGluonMM`).

Pixtral fine-tuned. In addition to the zero-shot and few-shot variants of Pixtral, we *instruct-finetuned* the model on question-answer pairs from our dataset in order to adapt the model to our domain (`Pixtral-ft`). We run LoRA fine-tuning for one epoch on 10,000 samples from the training set with a fixed learning rate of 3×10^{-5}.

4.2.4. With Training and With Reference Images

Finally, we study whether access to both training data and reference images improves performance.

PatchCore with a fine-tuned backbone. We test PatchCore as explained in Section 4.2.2, but replace the ResNet50 backbone fine-tuned on ImageNet with a ResNet50 backbone fine-tuned on our training dataset from Section 4.2.3 (`PatchCore50-ft`) similar to [16].

AutoGluonMM. To handle both query and reference images of the same item at train and test time, we use AutoGluon MultiModal [29] by passing all images (query *and* references) for each sample through the same image backbone and averaging their respective embeddings to obtain the final representation (`AutoGluon-ref`).

4.3. Results

We summarize the results of all baseline methods in Tables 2 and 3, with a detailed error analysis provided in the Supplementary Material IV. Our experiments aim to answer the following questions. (1) How well do methods with access to (all) defective instances at training time perform? (2) How does performance deteriorate when fewer defective instances are available for training? (3) How well do unsupervised and anomaly detection methods without access to defective instances for training perform?

(1) Upper bound with access to a large number of defective instances at training time. The supervised baselines perform well, with ViT-S reaching 90.67% $\mathbf{AP}_{\mathbf{any}}$. While models effectively detect major defects like deconstructions, penetrations, and deformations, they struggle with subtle anomalies, rare defect types (spillage), and reference-dependent defects (missing unit). False positives primarily occur with oddly-shaped items and "adversarial" items featuring damage-like designs. Notably, methods using both training data and references (PatchCore50-ft and AutoGluonMM-ref) underperform compared to reference-free approaches. This suggests that naive reference usage actually hinders model performance, likely due to feature averaging across input images complicating the learning task. This hypothesis is supported by inspecting their training set performance (96% $\mathbf{AP}_{\mathbf{any}}$ without references versus 87% with).

(2) Reduced access to defective instances at training time. Table 3 shows how supervised baselines perform in a more realistic scenario with limited number of defective training samples, with only 1% defective rate in the training set. Unsurprisingly, performance drops significantly from 90.67% $\mathbf{AP}_{\mathbf{any}}$ to 57.7% $\mathbf{AP}_{\mathbf{any}}$ for fully supervised methods (Query only). As before, the model is not able to leverage the non-defective samples (Query + ref).

Input	$\mathbf{AP}_{\mathbf{any}}$ [%]	$\mathbf{AP}_{\mathbf{major}}$ [%]	**AUROC**
Query only	57.7	40.5	74.4
Query + ref	40.4	14.9	63.2

Table 3. Classification performance on a reduced training set, with a defect rate of only 1%. We compare a ViT using only query images and a late-fusion ViT using both query *and* reference images.

(3) No defective instances at training time. No method without training surpasses 36.57% $\mathbf{AP}_{\mathbf{any}}$ (Claude-icl), with both zero-shot/in-context learning models (CLIP/POMP, Pixtral-⋆) and anomaly detection models (PatchCore50, WinCLIP-⋆) performing only slightly above chance. Out of the CLIP-based approaches the original CLIP model performs best. We find that the model seems to occasionally read the text on the items and wrongly associates it with defect predictions. VLMs provide a reasonable overall description and can catch egregious defects like gross deconstruction, but fail to capture the intricacy and variety of minor defects concerning deformable items, stickers/dirt on trays, and subtle anomalies, corroborating previous findings by [14]. Anomaly detection methods latch on to non-defect related visual differences, such as novel poses/viewpoints, background noise, and packaging variations.

Interestingly, PatchCore with a fine-tuned ResNet50 backbone (PatchCore50-ft) shows 4.32 ppts improvement compared to the ImageNet-based model PatchCore50, indicating the usefulness of leveraging defective instances for representation learning in anomaly detection [16]. However, it still struggles in detecting minor actuation and deconstruction, particularly when items are slightly displaced of their packaging. Moreover, some false negatives stem from faulty reference images incorrectly assumed to be non-defective, highlighting the extra pre-caution required in using unlabeled reference data in anomaly detection. False positives mostly arise from visual disparities between test and reference images, including variations in pose and product appearance. These results highlight the need for improved anomaly detection methods with a more thorough understanding of defects and more sophisticated ways for using references for visual comparison.

In summary, supervised methods perform best when given access to large amounts of defective instances during training, but still struggle with edge cases such as deformable and adversarial items. Adding reference images naively degrades rather than improves performance. Unsupervised and anomaly detection methods fall short by a significant margin, but improve with access to training data.

5. Outlook and Conclusion

We presented a large-scale dataset for visual defect and anomaly detection in retail logistics. Comprising 238,421 images including 29,316 defective samples, it captures challenges of retail logistics processes and represents one of the largest and most diverse datasets of its kind. The dataset overcomes critical limitations in existing benchmarks and enables the research community to address the remaining challenges in visual defect and anomaly detection. It allows for benchmarking methods in various scenarios, with and without training and reference images. We demonstrate the complexity of the proposed task by evaluating a number of state-of-the-art approaches and highlight the need for more robust solutions, particularly in anomaly-detection settings.

This dataset marks a significant step towards developing defect detection systems capable of handling real-world scenarios, setting a new standard for research in retail logistics applications of visual inspection. We encourage future research to explore the dataset by developing novel approaches. Key questions for future work include but are not limited to: (1) How can anomaly detection methods be generalized to deal with significant item and pose variability? (2) How can methods effectively leverage both training data and reference images? (3) How can we create methods that not only detect defects but also explain their reasoning?

Acknowledgements

We thank our collaborators in Amazon's operations, hardware and software engineering, as well as our annotation teams. Their invaluable contributions to hardware development, software implementation, data collection, and labeling efforts were essential to the success of this work.

References

[1] Pravesh Agrawal, Szymon Antoniak, Emma Bou Hanna, Baptiste Bout, Devendra Chaplot, Jessica Chudnovsky, Diogo Costa, Baudouin De Monicault, Saurabh Garg, Theophile Gervet, Soham Ghosh, Amélie Héliou, Paul Jacob, Albert Q. Jiang, Kartik Khandelwal, Timothée Lacroix, Guillaume Lample, Diego Las Casas, Thibaut Lavril, Teven Le Scao, Andy Lo, William Marshall, Louis Martin, Arthur Mensch, Pavankumar Muddireddy, Valera Nemychnikova, Marie Pellat, Patrick Von Platen, Nikhil Raghuraman, Baptiste Rozière, Alexandre Sablayrolles, Lucile Saulnier, Romain Sauvestre, Wendy Shang, Roman Soletskyi, Lawrence Stewart, Pierre Stock, Joachim Studnia, Sandeep Subramanian, Sagar Vaze, Thomas Wang, and Sophia Yang. Pixtral 12b, 2024. 2, 3, 6

[2] Samet Akcay, Amir Atapour-Abarghouei, and Toby P Breckon. Ganomaly: Semi-supervised anomaly detection via adversarial training. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14*, pages 622–637. Springer, 2019. 3

[3] Samet Akcay, Dick Ameln, Ashwin Vaidya, Barath Lakshmanan, Nilesh Ahuja, and Utku Genc. Anomalib: A deep learning library for anomaly detection. In *ICIP*, pages 1706–1710. IEEE, 2022. 3, 6

[4] Anthropic. The claude 3 model family: Opus, sonnet, haiku, 2024. 2, 3, 6

[5] Paul Bergmann, Sindy Löwe, Michael Fauser, David Sattlegger, and Carsten Steger. Improving unsupervised defect segmentation by applying structural similarity to autoencoders. *arXiv preprint arXiv:1807.02011*, 2018. 3

[6] Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, and Carsten Steger. The MVTec Anomaly Detection Dataset: A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. *International Journal of Computer Vision*, 129(4):1038–1059, 2021. 1, 2, 3, 4

[7] Qiyu Chen, Huiyuan Luo, Chengkan Lv, and Zhengtao Zhang. A unified anomaly synthesis strategy with gradient ascent for industrial anomaly detection and localization. *arXiv preprint arXiv:2407.09359*, 2024. 1

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 7

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 7, 6

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2, 6

[11] Songqiao Han, Xiyang Hu, Hailiang Huang, Minqi Jiang, and Yue Zhao. Adbench: Anomaly detection benchmark. *Advances in Neural Information Processing Systems*, 35: 32142–32159, 2022. 3

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2, 6, 7

[13] Jongheon Jeong, Yang Zou, Taewan Kim, Dongqing Zhang, Avinash Ravichandran, and Onkar Dabeer. Winclip: Zero-/few-shot anomaly classification and segmentation. In *CVPR*, pages 19606–19616, 2023. 2, 3, 6

[14] Xi Jiang, Jian Li, Hanqiu Deng, Yong Liu, Bin-Bin Gao, Yifeng Zhou, Jialin Li, Chengjie Wang, and Feng Zheng. MMAD: A Comprehensive Benchmark for Multimodal Large Language Models in Industrial Anomaly Detection, 2025. 3, 8

[15] Zishang Kong, Min He, Qianjiang Luo, Xiansong Huang, Pengxu Wei, Yalu Cheng, Luyang Chen, Yongsheng Liang, Yanchang Lu, Xi Li, and Jie Chen. Multi-Task Classification and Segmentation for Explicable Capsule Endoscopy Diagnostics. *Frontiers in Molecular Biosciences*, 8, 2021. 2, 3

[16] Mykhailo Koshil, Tilman Wegener, Detlef Mentrup, Simone Frintrop, and Christian Wilms. Anomalouspatchcore: Exploring the use of anomalous samples in industrial anomaly detection, 2024. 7, 8

[17] Pankaj Mishra, Riccardo Verk, Daniele Fornasier, Claudio Piciarelli, and Gian Luca Foresti. VT-ADL: A vision transformer network for image anomaly detection and localization. In *30th IEEE/IES International Symposium on Industrial Electronics (ISIE)*, 2021. 3

[18] Chaitanya Mitash, Fan Wang, Shiyang Lu, Vikedo Terhuja, Tyler Garaas, Felipe Polido, and Manikantan Nambi. ARMBench: An object-centric benchmark dataset for robotic manipulation. In *ICRA*, 2023. 3

[19] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. 7, 6

[20] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021. 6

[21] Shuhuai Ren, Aston Zhang, Yi Zhu, Shuai Zhang, Shuai Zheng, Mu Li, Alexander J Smola, and Xu Sun. Prompt pre-

training with twenty-thousand classes for open-vocabulary visual recognition. *Advances in Neural Information Processing Systems*, 36:12569–12588, 2023. 6

[22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. 4

[23] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14318–14328, 2022. 2, 3, 6

[24] Juan Jose Rubio, Takahiro Kashiwa, Teera Laiteerapong, Wenlong Deng, Kohei Nagai, Sergio Escalera, Kotaro Nakayama, Yutaka Matsuo, and Helmut Prendinger. Multi-class structural damage segmentation using fully convolutional networks. *Computers in Industry*, 112:103121, 2019. 2, 3

[25] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International conference on machine learning*, pages 4393–4402. PMLR, 2018. 3

[26] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Georg Langs, and Ursula Schmidt-Erfurth. f-anogan: Fast unsupervised anomaly detection with generative adversarial networks. *Medical image analysis*, 54:30–44, 2019. 3

[27] Jiyuan Shi, Ji Dang, Mida Cui, Rongzhi Zuo, Kazuhiro Shimizu, Akira Tsunoda, and Yasuhiro Suzuki. Improvement of Damage Segmentation Based on Pixel-Level Data Balance Using VGG-Unet. *Applied Sciences*, 11:pp.518.1–17, 2021. 2, 3

[28] Domen Tabernik, Samo Šela, Jure Skvarč, and Danijel Skočaj. Segmentation-Based Deep-Learning Approach for Surface-Defect Detection. *Journal of Intelligent Manufacturing*, 2019. 3

[29] Zhiqiang Tang, Haoyang Fang, Su Zhou, Taojiannan Yang, Zihan Zhong, Tony Hu, Katrin Kirchhoff, and George Karypis. Autogluon-multimodal (automm): Supercharging multimodal automl with foundation models. *arXiv preprint arXiv:2404.16233*, 2024. 7

[30] Hang Yao, Ming Liu, Haolin Wang, Zhicun Yin, Zifei Yan, Xiaopeng Hong, and Wangmeng Zuo. Glad: Towards better reconstruction with global and local adaptive diffusion models for unsupervised anomaly detection, 2024. 1, 2, 3

[31] Jihun Yi and Sungroh Yoon. Patch svdd: Patch-level svdd for anomaly detection and segmentation. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 3, 6

[32] Qinghui Zhang, Xianing Chang, and Shanfeng Bian Bian. Vehicle-Damage-Detection Segmentation Algorithm Based on Improved Mask RCNN. *IEEE Access*, 8:6997–7004, 2020. Conference Name: IEEE Access. 2, 3

[33] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In *ECCV*, 2022. 1, 2, 3, 4