Anomaly Detection for Cluttered, Multi-Instance E-grocery Stock Containers

Django Jagt

Delft University of Technology

TU Delft

Anomaly Detection for Cluttered, Multi-Instance E-grocery Stock Containers

by

Django Jagt

to obtain the degree of Master of Science
at the Delft University of Technology
to be defended publicly on May 28, 2026 at 13:00

Student number:	5052327	
Project duration:	September 01, 2025 – May 28, 2026	
Thesis committee:	Dr. rer. nat. M. Popović	TU Delft, Responsible supervisor
	Dr. ir. C. de Wagter	TU Delft, Chair
	Dr. J.C. van Gemert	TU Delft, External examiner
	Ir. K. Schultinga	Picnic Technologies, Supervisor

Cover: Generated using ChatGPT, personal prompt

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Preface

This thesis addresses a practical inspection problem: teaching a computer to spot leakage, open packaging, or the wrong product in a stock container, without first showing it what every quality issue looks like. The catch, as it turned out, is that the system can be equally enthusiastic about flagging harmless packaging clutter, stickers, cardboard trays, and the occasional stray leaf. The result is a pipeline that is equal parts anomaly detector and false-positive negotiator.

What the pipeline conveniently leaves out is the time spent looking at thousands of images. Apparently, building a system that detects anomalies from normal examples still requires a human to verify what is normal, what is not, and what is simply a piece of paper in an unfortunate position.

Thank you to my supervisors, Marija Popović and Kevin Schultinga, for their guidance, critical feedback, and support throughout this project. I would also like to thank Picnic Technologies for providing the real-world data and operational context that kept the research grounded. Finally, thank you to the DelftBlue cluster for generating more anomaly maps than I care to count.

Here's to teaching machines not only what to detect, but also what to ignore.

Django Jagt
Delft, May 2026

Contents

Nomenclature

Abbreviation	Definition
CLAHE	Contrast-limited adaptive histogram equalisation
CNN	Convolutional neural network
DINOv2	Self-distillation with no labels, version 2
FAISS	Facebook AI Similarity Search
F1	Harmonic mean of precision and recall
FN	False negative
FP	False positive
FPS	Farthest point sampling
HSV	Hue, saturation, value
IoU	Intersection over union
mNND	Mean nearest-neighbour distance
OCR	Optical character recognition
RGB	Red, green, blue
ROI	Region of interest
SKU	Stock keeping unit
TN	True negative
TP	True positive
ViT	Vision Transformer
VLM	Vision-language model

Symbol	Definition
I_q	Query image
s	Expected SKU associated with the query image
$\mathcal{P}_s$	Clean reference pool for SKU s
$\mathcal{R}_s$	Sampled SKU-specific reference set used during inference
$\mathcal{V}_s$	SKU-specific reference vocabulary used for variant verification
$\mathcal{M}_s$	SKU-specific reference memory bank of patch embeddings
M	Number of normal images in $\mathcal{R}_s$
M_s	Number of images in $\mathcal{P}_s$
N	Target reference-set size used during reference sampling
$N_{\min}$	Minimum number of observations required for container-level filtering
P	Number of extracted patches in an image
D	Patch-feature embedding dimension
$f(\cdot)$	Visual feature extractor
$\mathbf{p}_i$	Patch-level feature embedding
$d(\cdot,\cdot)$	Cosine distance between feature embeddings
$d_{\mathrm{NN}}(\cdot;\mathcal{M}_s)$	Nearest-neighbour distance to $\mathcal{M}_s$
a_i	Patch-level anomaly score for patch i
$A(I_q)$	Patch-level anomaly map for query image I_q
$S(I_q)$	Image-level anomaly score for query image I_q
$q(\cdot)$	Aggregation function for image-level scoring
$H_{0.01}(\cdot)$	Subset containing the top 1% largest patch-level anomaly scores
$W(I_q)$	Edge-based weight map used for edge downweighting
$w_{\min}$	Minimum edge weight used during edge downweighting
$\tilde{A}(I_q)$	Reweighted anomaly map after edge downweighting
$\mathcal{C}(I_q)$	Set of candidate crops extracted from query image I_q
$\hat{y}_c$	Binary debris-filter prediction for crop c
$\hat{y}_{\mathrm{df}}(I_q)$	Image-level debris-filter decision
$\mathcal{T}(I_q)$	Set of OCR tokens detected in query image I_q
$\alpha(t)$	Indicator that token t mismatches the reference vocabulary
$\hat{y}_{\mathrm{vv}}(I_q)$	Image-level variant verification decision
$\mathcal{C}_m$	Set of reference images assigned to cluster m
$\boldsymbol{\mu}_m$	Centroid of cluster m
r_m	Radius of cluster m

Part I

Preliminary

Introduction

The shift towards online grocery shopping has increased the need for reliable automated quality inspection in fulfilment operations. E-grocery fulfilment centres process thousands of products daily under strict throughput constraints, and quality issues including wrong products, product damage, packaging damage, and leakage must be identified before goods reach the customer. Unlike conventional retail, where customers perform a final check at the shelf, the fulfilment centre represents the last opportunity to detect such issues before delivery. Automated visual inspection offers a direct route to improve detection reliability without adding substantial manual inspection overhead.

Realising this in practice is non-trivial. In a goods-to-person fulfilment setting, each stock container is captured by a fixed overhead RGB camera as it passes along the conveyor. The resulting image contains multiple partially overlapping product instances with varying pose, orientation, and visibility. Container wear, stickers, reflections, loose transport material, and packaging contours can all produce local image regions that resemble genuine quality issues. Any inspection system must therefore be sensitive to real defects while remaining robust to these benign nuisance sources, across a large and continuously changing product assortment, and within a strict operational latency budget.

Existing industrial anomaly detection methods do not fully address this combination of requirements. Standard benchmarks primarily represent controlled, single-object inspection scenarios and provide limited evidence for performance under dense clutter and large within-class appearance variation. Few-shot and reference-based methods reduce the need for defective training examples and support large product assortments, but their performance remains sensitive to reference quality and scene complexity. This thesis addresses the resulting gap by developing a modular reference-based inspection pipeline in which stock keeping unit (SKU)-specific normality is modelled through automatically constructed and sampled reference sets, false positives are suppressed through targeted mitigation stages, and the complete system remains feasible under warehouse deployment constraints. The research questions guiding this investigation are formulated in the following chapter.

Research Questions

This thesis addresses automated quality inspection of cluttered, multi-instance e-grocery stock containers, where each container is associated with a known stock keeping unit (SKU). The inspection setting requires visible product and packaging anomalies to be detected from single overhead RGB images, while limiting false positives caused by benign clutter and normal appearance variation. Based on this problem setting and the reviewed anomaly detection literature, the thesis is guided by the following research objective and research questions.

Research objective The objective of this research is to design and evaluate a modular reference-based anomaly detection pipeline for e-grocery stock-container inspection. The pipeline should detect visible wrong products, product damage, packaging damage, and leakage while limiting false positives caused by benign clutter and remaining feasible under warehouse deployment constraints.

Main research question

> **How can a reference-based anomaly detection pipeline be designed to detect visible anomalies in cluttered, multi-instance e-grocery stock containers while limiting false positives and remaining scalable under warehouse deployment constraints?**

Research Question 1 How can a compact and reliable SKU-specific reference set be automatically constructed for reference-based anomaly detection across a large and changing e-grocery product assortment?

Research Question 2 How can false positives caused by product boundaries, packaging clutter, loose debris, and container artefacts be reduced while preserving sensitivity to genuine product and packaging anomalies?

Research Question 3 How can the complete inspection pipeline remain computationally feasible under warehouse latency constraints while maintaining detection reliability?

Together, these research questions define the scope of the thesis. The first question addresses the automatic construction of compact and reliable SKU-specific reference sets across a large and changing product assortment. The second question addresses robustness in cluttered multi-instance scenes, where high local anomaly responses do not necessarily correspond to operationally relevant defects. The third question addresses the practical requirement that the inspection pipeline remains feasible for warehouse deployment, where end-to-end latency constrains the design.

Part II

Scientific Article

Anomaly Detection for Cluttered, Multi-Instance E-grocery Stock Containers

Django Jagt

Abstract — **Automated quality inspection of cluttered, multi-instance e-grocery stock containers is challenging because packaging clutter, product boundaries, container wear, and loose debris generate local anomaly responses that do not correspond to genuine quality issues. This thesis develops a modular reference-based inspection pipeline in which normality for each stock keeping unit (SKU) is modelled through automatically constructed and sampled reference sets. Query images are scored by patch-level nearest-neighbour matching of DINOv2 features against a reference memory bank, following the AnomalyDINO paradigm. Edge downweighting and crop-level debris filtering suppress boundary- and debris-driven false positives, while an optical character recognition (OCR)-based branch additionally addresses visually subtle wrong-SKU substitutions. Evaluated on 2,996 normal and 1,657 issue images across 20 SKU classes and five issue families, the full pipeline increases precision from 0.511 to 0.834 and reduces false positives from 1,414 to 201 relative to the core detector (F1-score: 0.705). This improvement comes at a recall cost, with recall decreasing from 0.893 to 0.610. The full pipeline achieves a 99th-percentile end-to-end latency of 1.4 s, within the 10 s operational constraint. The results demonstrate that robust inspection in this setting requires treating false-positive mitigation as a core system requirement rather than relying on raw anomaly sensitivity alone.**

Keywords: visual anomaly detection, reference-based anomaly detection, e-grocery quality inspection, cluttered multi-instance scenes, DINOv2, AnomalyDINO, patch-level nearest-neighbour matching, false-positive mitigation

1 Introduction

The shift towards online grocery shopping has increased the need for reliable and scalable fulfilment operations. E-grocery fulfilment centres must process large numbers of products while maintaining high standards for order correctness, product quality, and throughput. Wrong, damaged, or leaking items can lead to complaints, compensation costs, waste, and additional rework within the fulfilment process. Unlike in conventional supermarkets, where customers can visually inspect products at the shelf, e-grocery systems must perform this quality assurance before products leave the fulfilment centre. Automated visual inspection therefore offers a direct opportunity to improve reliability without adding substantial manual inspection overhead.

Within this operational context, this thesis considers automated inspection of e-grocery stock containers in a goods-to-person fulfilment setting, where containers are transported to a picking station and a human operator transfers products into order containers. Each stock container is associated with a known expected stock keeping unit (SKU) and is captured by a fixed overhead RGB camera as it passes along the conveyor. The inspection task is to determine, from a single image, whether the container contains visible evidence of a wrong product, product damage, packaging damage, or leakage before it continues downstream. This setting is challenging because normal containers may contain many partially overlapping instances with varying pose, orientation, visibility, and packaging context. In addition, container wear, labels, reflections, transport material, and loose debris can produce high local anomaly responses despite not representing quality issues. The problem therefore requires a detector that is sensitive to local deviations while remaining robust to benign clutter and normal variation.

Existing industrial anomaly detection methods do not fully address this setting. Standard benchmarks such as MVTec AD and VisA primarily represent controlled inspection scenarios and provide limited evidence for performance in dense, multi-instance stock-container imagery [1, 2]. Few-shot and reference-based methods reduce the need for defective examples and per-SKU training, making them attractive for large SKU assortments, but they remain sensitive to reference quality and coverage of normal variation [3, 4]. In cluttered stock containers, raw patch-level anomaly scoring can therefore assign high scores to benign structure rather than true quality issues. Vision-language approaches may improve flexibility, but can introduce additional prompting and multimodal reasoning overhead that complicates latency-sensitive deployment [5]. This leaves a gap between current anomaly detection methods and robust, scalable inspection of cluttered e-grocery stock containers under real-time deployment constraints.

This thesis therefore addresses the following research question: **How can a reference-based anomaly detection pipeline be designed to detect visible anomalies in cluttered, multi-instance e-grocery stock containers while limiting false positives and remaining scalable under warehouse deployment constraints?**

To address this question, this thesis develops a modular reference-based inspection pipeline in which SKU-specific normality is modelled through automatically constructed and sampled reference sets. Query images are scored through patch-level nearest-neighbour matching of DINOv2 features against the corresponding reference memory bank, following the AnomalyDINO paradigm [4, 6]. Because raw patch-level scores are vulnerable to clutter-induced false positives, the core detector is augmented with edge down-weighting and debris filtering. An OCR-based variant verification branch further addresses visually subtle wrong-SKU cases. The resulting system is evaluated as a deployment-oriented anomaly detection pipeline rather than as a standalone detector, with emphasis on precision–recall trade-offs, reference-set design, false-positive mitigation, and runtime feasibility.

The main contributions of this thesis are:

1. A reference-set construction and sampling strategy for cluttered e-grocery inspection, showing that automated filtering and diversity-aware sampling can produce clean, compact SKU-specific reference sets suitable for scalable deployment.
2. A targeted boundary-mitigation approach that identifies normal product contours as a systematic false-positive source in cluttered multi-instance stock containers, and uses edge down-weighting to suppress boundary-driven nuisance responses.
3. A crop-level nuisance-filtering approach that treats high-scoring local anomaly regions as candidate evidence requiring validation, suppressing benign artefacts such as loose debris, labels, container wear, and transport material before image-level decision-making.
4. A controlled deployment-oriented evaluation of the complete inspection pipeline, demonstrating that the full design achieves a practical precision–recall operating point within warehouse latency constraints, and providing exploratory evidence that the reference-based scoring principle transfers to drone-based acquisition contexts.

The remainder of this thesis is structured as follows. Chapter 2, Related Work, reviews industrial visual anomaly detection, few-shot and reference-based anomaly detection, and anomaly detection in cluttered multi-instance scenes, and identifies the research gap. Chapter 3, Preliminaries, defines the inspection task, operational assumptions, and reference-based anomaly detection formulation. Chapter 4, Methodology, presents the proposed inspection pipeline. Chapter 5, Experiments, evaluates the pipeline through end-to-end performance, reference-set design, false-positive mitigation, runtime analysis, and a transferability experiment. Chapter 6, Discussion, interprets the design implications, limitations, and deployment scope of the evaluated pipeline. Chapter 7, Conclusion, summarises the findings and outlines directions for future work.

2 Related Work

2.1. Industrial Visual Anomaly Detection

Industrial visual anomaly detection addresses the identification of product defects or other deviations from normal appearance when defective training examples are scarce, incomplete, or unavailable [1, 3]. In industrial inspection, this has led to a shift from supervised defect classification towards methods that learn normality from anomaly-free data and flag deviations at inference time [1, 3, 7]. The literature can be grouped broadly into reconstruction-based, representation- and memory-based, flow-based, diffusion-based, and, more recently, vision–language or foundation-model-based approaches [3, 8–12]. These families differ mainly in how they represent normal appearance, how anomalies are scored, and how readily they support localisation [3, 9–11].

Reconstruction-based methods detect anomalies through discrepancies between an input and its reconstruction [8]. Representation- and memory-based methods instead compare deep feature embeddings of a query image against a learned or stored representation of normality [3, 13]. Flow-based methods instead estimate the likelihood of normal data in feature space [9, 14], while diffusion-based methods reconstruct anomaly-free counterparts through iterative denoising [10, 15]. More recent vision–language approaches use large pre-trained models to perform zero-shot or few-shot anomaly reasoning through multimodal representations [11, 12, 16]. Collectively, these developments have improved data efficiency and broadened applicability, but they also introduce different trade-offs in localisation quality, dependence on representative normal data, and computational cost [3, 4, 11, 17]. In particular, industrial settings often impose strict runtime constraints, making latency and deployability important considerations alongside detection performance [17].

For this thesis, the most relevant direction is the sub-

set of representation- and memory-based methods that define normality from a limited set of clean reference samples rather than through extensive per-class training [3, 4, 13]. This is motivated by two practical properties of the stock-container inspection setting: defective examples are scarce and heterogeneous, while the inspected SKU assortment is large and continuously changing. Under these conditions, methods that rely on normal data only and support efficient adaptation to new products are more suitable than approaches requiring extensive labelled data [3, 4] or computationally heavier generative inference [10, 15]. Few-shot and reference-based anomaly detection methods within this broader family are therefore discussed in more detail in the next section.

2.2. Few-Shot and Reference-Based Anomaly Detection

Few-shot and reference-based anomaly detection is particularly relevant when defective samples are scarce and per-class model training is impractical. In industrial inspection, this setting arises when new product categories must be onboarded with limited clean data and without reliable coverage of defect types. Rather than learning a separate anomaly model for each category, these methods define normality from a small set of clean reference samples and compare test observations against this reference representation at inference time. This makes them attractive for large and changing product assortments, where scalability depends on minimising annotation effort, retraining cost, and category-specific engineering [3, 4, 13].

The core mechanism of these methods is to represent normal appearance through a support set or memory bank, and to score deviations by comparing query features to this reference distribution. In patch-based methods, feature embeddings extracted from clean reference images are stored and later matched against patch embeddings from the query image. Anomaly scores are then derived from the distance between a query patch and its closest normal counterpart, often through nearest-neighbour matching. This patch-level comparison is well suited to localised defects, since it preserves spatial information and allows anomaly maps to be constructed directly from local feature discrepancies. Representative approaches such as DN2, SPADE, PatchCore, and AnomalyDINO follow this general paradigm, differing mainly in feature extraction, memory construction, and scoring design [3, 4, 13, 18]. Related unified few-shot methods such as UniVAD extend this idea with component-level reasoning, but also rely more strongly on consistent segmentation and incur higher computational cost [19].

Recent progress in few-shot anomaly detection has been driven in large part by stronger pre-trained visual representations. Earlier approaches commonly relied on supervised convolutional neural network (CNN) features, whereas more recent methods benefit from self-supervised foundation models that provide more transferable and semantically meaningful patch embeddings. In particular, DINOv2 has proven effective for patch-level anomaly detection because it yields robust representations without task-specific fine-tuning and retains sufficient local structure for nearest-neighbour matching and anomaly localisation. This explains the strong performance of AnomalyDINO, which combines the deep nearest-neighbour paradigm with DINOv2 features in a training-free inference pipeline [3, 4, 6].

Despite these advantages, performance in few-shot and reference-based anomaly detection remains strongly dependent on reference quality, coverage, and selection. If the reference set does not capture valid normal variation in pose, scale, illumination, or partial visibility, normal test patterns may be assigned high anomaly scores. Conversely, if anomalous or ambiguous samples enter the reference memory, true defects may be suppressed. This sensitivity becomes especially problematic in cluttered multi-instance scenes, where background context, overlap, and nuisance regions can affect local feature matching even when the product itself is normal. Consequently, although few-shot reference-based methods provide an effective foundation for scalable anomaly detection, their success in the stock container inspection setting depends not only on the detector itself but also on the construction, representativeness, and maintenance of the reference set [3, 4, 13].

2.3. Anomaly Detection in Cluttered Multi-Instance Scenes

Most industrial visual anomaly detection benchmarks and evaluations remain centred on relatively clean inspection settings in which one object, or only a small number of objects, is shown against a controlled background. Datasets such as MVTec AD and VisA have been instrumental in advancing the field, but they provide only limited evidence for robustness under dense clutter, strong overlap, and large scene variability [1, 2]. Even benchmarks that extend beyond simple surface defects, such as MVTec LOCO AD and KAPUTT, do not fully reflect the combination of multi-instance clutter, random arrangement, and nuisance background variation that characterises stock-container inspection [20, 21]. As a result, anomaly detection in cluttered multi-instance scenes should be regarded as a distinct problem regime rather than as a direct extension of standard industrial anomaly detection.

This distinction matters because overlap, partial visib-

ility, nuisance context, and large within-class appearance variation complicate local anomaly scoring and reduce the direct transferability of results obtained on cleaner benchmarks. Related datasets such as MVTec D2S and ARMBench capture important aspects of cluttered multi-object perception, but they do not constitute the same anomaly detection setting as real stock-container inspection [22, 23]. Consequently, the limitation is not merely the absence of a perfect benchmark. More fundamentally, existing anomaly detection literature still provides limited evidence for reliable performance under cluttered, multi-instance conditions with substantial normal variation.

2.4. Research Gap and Thesis Positioning

Recent advances in industrial visual anomaly detection have improved data efficiency and broadened the applicability of class-agnostic inspection methods. In particular, few-shot and reference-based approaches provide a practical alternative to fully supervised per-class training by defining normality from a small set of clean reference samples [3, 4]. However, the literature still leaves an important gap between benchmark progress and the practical requirements of cluttered stock-container inspection. Existing methods provide limited evidence for reliable performance when scenes contain multiple overlapping instances and deployment must remain computationally efficient across a large and changing SKU assortment [1, 2, 20, 21].

This gap is not explained by a single missing dataset or method component. Rather, it reflects the broader difficulty of applying current anomaly detection methods in settings with dense clutter, occlusion, and large within-class appearance variation. In the stock-container setting, anomaly detection must remain robust to normal changes in arrangement, visibility, and packaging context, while still responding to genuine product and packaging defects. At the same time, the reference construction and inference process must scale across thousands of products without introducing prohibitive memory, maintenance, or runtime costs.

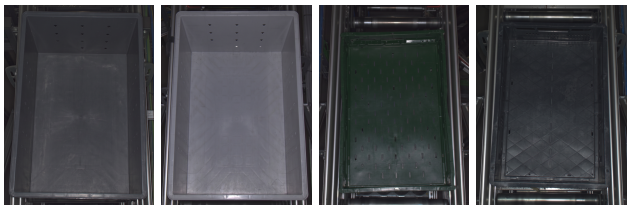
This thesis is positioned within the reference-based few-shot anomaly detection literature and adopts patch-level nearest-neighbour matching as its methodological foundation. More specifically, it builds on the computationally practical and training-free direction represented by AnomalyDINO, and adapts this reference-based perspective to the requirements of cluttered multi-instance stock-container scenes [4]. The proposed framework therefore focuses on three connected objectives: constructing a robust SKU-specific reference pool and sampling from it to obtain a representative reference set, improving reliability in cluttered scenes through targeted false-positive mitigation, and maintaining practical deployability through modular and latency-aware design. In this way, the thesis does not propose a completely new anomaly detection paradigm, but a problem-adapted extension of reference-based anomaly detection for automated stock-container inspection.

3 Preliminaries

This chapter defines the problem setting, operational assumptions, and formal reference-based anomaly detection formulation underlying the proposed method.

3.1. Problem Setting

This thesis considers automated quality inspection of e-grocery stock containers in a goods-to-person fulfilment setting. In the operational context considered here, stock containers are rigid reusable plastic containers used for product storage and transport within the automated fulfilment system. Two stock-container types are considered, namely stock totes and stock crates (Figure 3.1). Unless a distinction is required, stock totes and stock crates are referred to jointly as *stock containers* throughout this thesis.



(a) (b) (c) (d)

Figure 3.1: Example stock-container types used in the operational setting: (a) ambient tote, (b) chilled tote, (c) green crate, and (d) black crate. Unless stated otherwise, these are referred to jointly as *stock containers*.

At inference time, a query image $I_q \in \mathbb{R}^{H \times W \times 3}$ is captured by an overhead RGB inspection camera and corresponds to one expected stock keeping unit (SKU) s. The image shows a stock container with multiple visible instances of that product. Given I_q and a SKU-specific reference set $\mathcal{R}_s$ of anomaly-free images, the inspection task is to determine whether the image contains visible evidence of a product- or packaging-related quality issue. When such evidence is present, the system should further indicate the corresponding image region to support localisation and downstream inspection handling.

3.2. Operational Assumptions

The following assumptions define the intended operating conditions under which the proposed framework is designed and evaluated.

- **Known expected SKU:** The expected SKU s is assumed to be known at inference time through

correct barcode reading and correct image-to-barcode association.

- **Single-SKU expectation:** Each stock container is assumed to contain instances of one expected SKU only. The presence of a different SKU or a mixture of SKUs is considered anomalous.
- **Image-wise inference:** Although multiple images of the same stock container may be captured while it moves through the conveyor system, the inspection pipeline processes each image independently.
- **Image acquisition:** Images are acquired using fixed RGB cameras mounted above the conveyor belt at a shallow viewing angle of approximately 12°. Small variations in lighting and camera placement may occur across camera stations, but the imaging setup is otherwise consistent.
- **Reference availability:** A SKU-specific reference set $\mathcal{R}_s$ is assumed to be available for each inspected product. These references are intended to represent clean, anomaly-free product appearance and remain fixed during inference.
- **Scene nuisance factors:** Container wear and incidental non-product artefacts, such as stickers or loose transport material, may be present in the scene. Products may further overlap, partially occlude one another, or be stacked.
- **Visible-evidence limitation:** The framework is only expected to detect issues that are visually observable in the acquired image. Defects that are fully occluded or otherwise not visible fall outside the scope of the image-based inspection task.
- **Operational latency:** The final inspection decision is required within approximately 10 s after image acquisition in order to support downstream rework handling before the stock container returns to storage.

These assumptions bound the problem setting considered in this thesis and define the operational conditions under which the proposed anomaly detection pipeline is intended to operate.

3.3. Reference-Based Anomaly Detection Formulation

Let $I_q \in \mathbb{R}^{H\times W\times 3}$ denote a query image of SKU s, and let $\mathcal{R}_s = \{I_r^{(1)}, \ldots, I_r^{(M)}\}$ denote a SKU-specific reference set containing M normal images of the same SKU. A feature extractor f maps each image to a tuple of patch-level feature embeddings:

$$f(I) = (\mathbf{p}_1, \ldots, \mathbf{p}_P), \qquad \mathbf{p}_i \in \mathbb{R}^D,$$

where P denotes the number of extracted patches, D the embedding dimension, and $[P] = \{1, \ldots, P\}$.

The normal reference memory bank for SKU s is defined as the collection of all patch embeddings extracted from the reference set:

$$\mathcal{M}_s = \bigcup_{m=1}^{M} \left\{ \mathbf{p}_j^{(m)} \;\middle|\; j \in [P], \; f(I_r^{(m)}) = (\mathbf{p}_1^{(m)}, \ldots, \mathbf{p}_P^{(m)}) \right\}.$$

Given a query image I_q, anomaly detection is performed by comparing each query patch embedding $\mathbf{p}_i^q \in f(I_q)$ to its nearest neighbour in $\mathcal{M}_s$. Following *AnomalyDINO* [4], the patch-level anomaly score is defined as

$$a_i = d_{\mathrm{NN}}(\mathbf{p}_i^q; \mathcal{M}_s) = \min_{\mathbf{p}\in\mathcal{M}_s} d(\mathbf{p}_i^q, \mathbf{p}),$$

where $d(\cdot,\cdot)$ is the cosine distance

$$d(\mathbf{x}, \mathbf{y}) = 1 - \frac{\langle \mathbf{x}, \mathbf{y}\rangle}{\|\mathbf{x}\|\,\|\mathbf{y}\|}.$$

Collecting the patch-level scores yields an anomaly map

$$A(I_q) = (a_1, \ldots, a_P),$$

which indicates the spatial distribution of local deviations from the normal reference distribution. The corresponding image-level anomaly score is obtained by aggregating the patch-level distances,

$$S(I_q) = q(A(I_q)),$$

where, following *AnomalyDINO*, $q(\cdot)$ is defined as the mean of the 1% highest patch-level anomaly scores,

$$q(\mathcal{D}) = \mathrm{mean}(H_{0.01}(\mathcal{D}))\,,$$

with $H_{0.01}(\mathcal{D})$ denoting the subset containing the top 1% largest values in $\mathcal{D}$. The image-level score $S(I_q)$ and the corresponding anomaly map $A(I_q)$ form the raw anomaly-scoring outputs used by the downstream mitigation and decision stages.

4 Methodology

This chapter presents the proposed reference-based inspection pipeline as a modular system consisting of an offline SKU-specific reference system and three inference-time branches: anomaly detection, false-positive mitigation, and variant verification. The reference system constructs clean, compact reference sets and a reference vocabulary for each SKU. At inference time, the anomaly detection branch produces local anomaly responses, the false-positive mitigation branch

suppresses nuisance responses from benign clutter and container-related artefacts, and the variant verification branch uses OCR-based vocabulary matching to detect subtle wrong-variant cases. The chapter concludes with computational efficiency and deployment considerations, focusing on latency, memory-bank size, conditional execution, and runtime trade-offs.

4.1. System Overview

Figure 4.1 summarises the proposed inspection pipeline. The system consists of a SKU-specific reference system and an inference branch. The reference system constructs a clean reference pool $\mathcal{P}_s$ from candidate stock-container images, samples from this pool to obtain the compact inference-time reference set $\mathcal{R}_s$, and derives the reference vocabulary $\mathcal{V}_s$ used for variant verification. These components provide the visual and text-level reference support used during inference.

The inference branch operates on each query image I_q. The anomaly detection branch applies image preprocessing and generates a patch-level anomaly map $A(I_q)$ together with an image-level anomaly score $S(I_q)$. The false-positive mitigation branch then refines this raw anomaly evidence through product boundary extraction, edge downweighting, candidate crop extraction, and debris filtering. This suppresses nuisance responses caused by normal product-boundary transitions, container wear, loose transport material, and other benign clutter. In parallel, the variant verification branch extracts OCR tokens from the query image and compares them with the SKU-specific reference vocabulary to detect subtle wrong-variant cases. The debris-filtered anomaly evidence and the variant verification output are combined in the final inspection decision. The following sections describe each component in detail.

4.2. Reference System

The reference system, highlighted as the green section in Figure 4.1, constructs the SKU-specific reference components that support inference: the clean reference pool $\mathcal{P}_s$, the compact reference set $\mathcal{R}_s$ sampled from this pool, and the reference vocabulary $\mathcal{V}_s$. Together, these components provide the visual and text-level reference support used to compare each query image with the expected SKU.

4.2.1. Reference Pool Construction

Motivation - Reference-based anomaly detection requires a clean SKU-specific reference set $\mathcal{R}_s$ that captures normal product appearance without including defective samples. In the proposed pipeline, this inference-time reference set is sampled from a larger clean reference pool $\mathcal{P}_s$ constructed through automated filtering. Manual construction of this pool is infeasible in the present e-grocery setting because the product assortment is large and continuously chan-

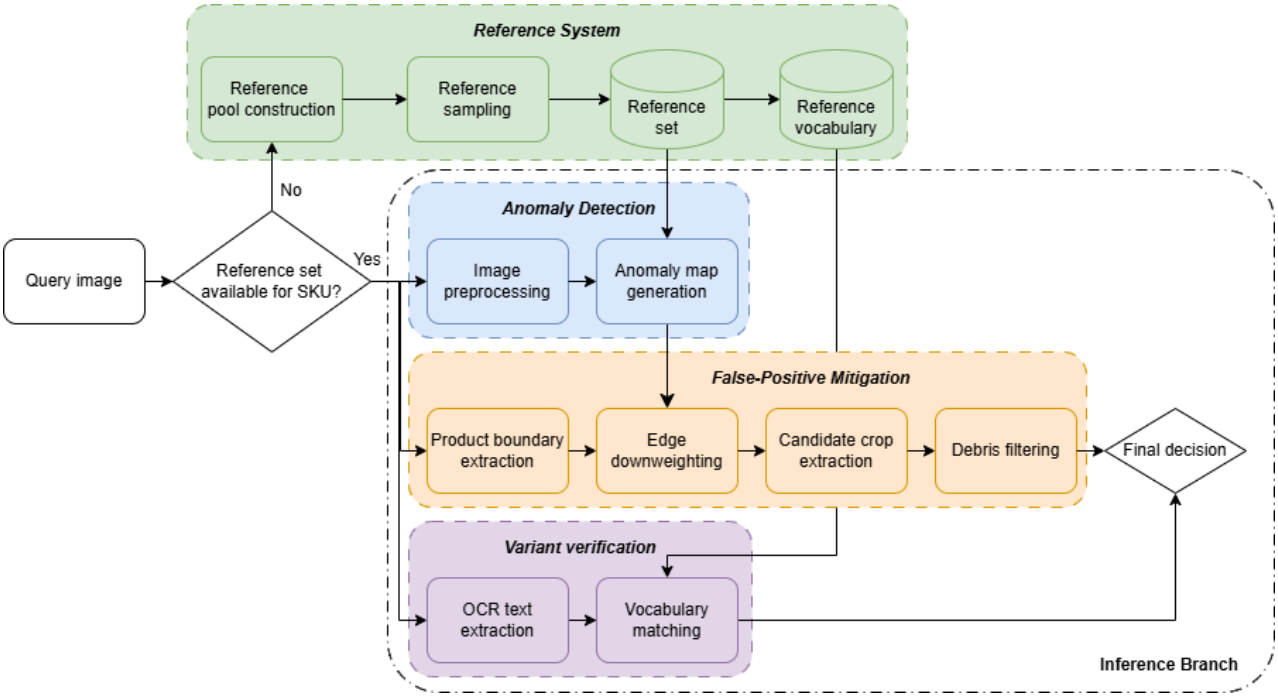


Figure 4.1: Overview of the proposed reference-based inspection pipeline. The SKU-specific reference system constructs the clean reference pool, compact reference set, and reference vocabulary. During inference, the query image is processed by the anomaly detection branch, the false-positive mitigation branch, and the variant verification branch before the final inspection decision is made.

ging. Automatic selection is also non-trivial, since normal observations vary in pose, orientation, instance count, and packaging context, while subtle defects such as leakage or packaging damage may be difficult to identify reliably from a single image. Reference construction is therefore formulated as a high-precision filtering problem in which uncertain samples are rejected conservatively, favouring a clean reference pool over size.

Pipeline Overview - The automated construction of the reference pool $\mathcal{P}_s$ follows a two-stage filtering pipeline, illustrated in Figure 4.2: (i) image-level automatic labelling of candidate images, followed by (ii) container-level consistency filtering across all observations of the same container–SKU combination. A vision-language model (VLM) is used in the first stage because it supports zero-shot image-level labelling across a large and continuously changing SKU assortment without requiring per-SKU supervised training. Its decision is guided using SKU-specific template images and product metadata. To avoid unnecessary processing, the pipeline is applied only to SKUs whose current reference pool remains below fixed operational thresholds for both the number of accepted images and the number of distinct containers. These two criteria control not only the total amount of reference data, but also its diversity across container instances.

Stage 1: Image-Level Automatic Labelling - In the first stage, each candidate image is evaluated independently for inclusion in the reference pool. The vision-language model receives the cropped container image together with the expected SKU template image and associated product metadata, and predicts whether the observation is *normal* or *anomalous*. This image-level label is then used as the input to the subsequent container-level consistency filter.

To improve robustness across the heterogeneous SKU assortment, the prompt is adapted dynamically to the expected product. This allows the model to apply product-specific decision criteria rather than relying on a generic visual assessment alone. For leakage-sensitive products, an additional leakage inspection is performed on a zoomed-in view of the container floor. An image is classified as anomalous if either the general inspection or the leakage-specific inspection indicates a defect. The VLM routing, metadata fields, and representative prompt structure used for this filtering stage are documented in Appendix D.

Stage 2: Container-Level Consistency Filtering - Image-level predictions are aggregated at container level by grouping all observations of the same container–SKU combination. If any image in the group is classified as anomalous, the entire group is excluded from the clean reference pool. This conservative rule mitigates occasional errors in image-level labelling, where subtle defects may be missed or incorrectly classified. In addition, only groups containing at least $N_{\min}$ images are considered, where $N_{\min}$ denotes the minimum required number of observations per container–SKU group. This ensures that container-level decisions are based on sufficient observations and reduces the likelihood of including samples affected by image-level false-negative errors.

Final Selection Rule - The proposed construction strategy prioritises precision over recall by discarding uncertain samples to minimise reference contamination. An image is included in the final reference pool only if it is classified as normal at image level and its container–SKU group passes the container-level consistency filter. In practice, consistently low acceptance rates for a given SKU are monitored, as they may indicate filtering bias or issues with the template image or metadata. Representative cases in which valid normal appearances may be excluded are shown in Figure 4.3.

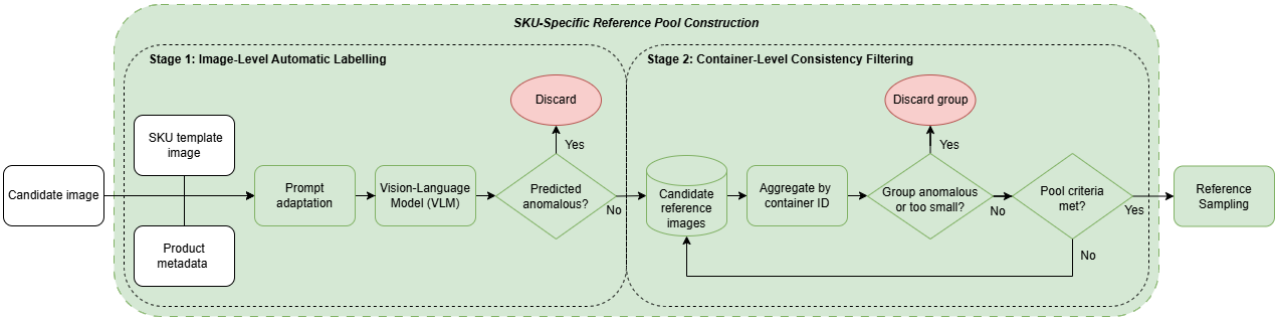


Figure 4.2: Overview of the automated SKU-specific reference-pool construction pipeline. In Stage 1, candidate images are labelled at image level using the expected SKU template image and product metadata. In Stage 2, image-level labels are aggregated and filtered conservatively at container level before reference sampling is applied.

(a) (b) (c) (d)

Figure 4.3: Representative ambiguity cases during automated reference-pool construction. Each column shows a SKU template image in the top row and a candidate product crop extracted from a stock-container image in the bottom row. (a) A transparent upside-down product may resemble leaked or spilled content. (b) The template and operational product appearance may differ. (c) Secondary packaging text or advertisements may be confused with a wrong product. (d) A valid package orientation may resemble opened packaging. These cases illustrate a limitation of conservative VLM-based filtering: valid normal appearances may be excluded from the reference pool when they are visually ambiguous or differ from the template appearance.

4.2.2. Reference Sampling

Motivation - Following the construction of a clean SKU-specific reference pool $\mathcal{P}_s$ (Subsection 4.2.1), a sampling step is applied to obtain a compact yet representative reference set $\mathcal{R}_s \subset \mathcal{P}_s$ for each SKU. The objective is to reduce the size of the reference data while preserving the diversity of valid product appearances, including variations in orientation, spatial arrangement, and packaging context. This is important for practical deployment because large and redundant reference sets increase the computational cost of nearest-neighbour matching. In addition, excessive redundancy can reduce separability between normal and anomalous query images. The effect of reference subset size on anomaly detection performance is evaluated separately in Section 5.3.

Although anomaly detection is performed at patch level, reference sampling is performed at image level because each selected image contributes its full set of patch embeddings to the downstream memory bank. Image-level diversity therefore acts as a practical proxy for patch-level coverage. Selecting visually distinct normal images introduces varied local product, packaging, and context patterns, while avoiding redundant images limits repeated patch features. The sampling step is therefore a compactness and coverage mechanism for the reference memory bank.

Sampling in Feature Space - To maintain consistency with the downstream anomaly map generation, reference sampling uses the same DINOv2-based backbone [4, 6]. Let $\mathcal{P}_s = \{I^{(1)}, \ldots, I^{(M_s)}\}$ denote the clean reference pool for SKU s, where $M_s = |\mathcal{P}_s|$. Each reference image is encoded into an image-level embedding $\mathbf{f}^{(i)} \in \mathbb{R}^d$, where d denotes the embedding dimension. The embeddings are ℓ_2-normalised and compared using the cosine distance $d(\cdot,\cdot)$ defined in Section 3.3, such that visually similar samples lie close to one another in feature space.

Given a target reference-set size N, sampling is performed using k-means clustering with boundary top-up. First, the reference pool is partitioned into

$$k = \lceil 0.67\, N \rceil$$

clusters in feature space using k-means on the normalised embeddings. This allocation prioritises representative coverage of dominant normal modes while reserving a fixed portion of the budget for boundary top-up samples that support less frequent but valid normal appearances. Let $\mathcal{C}_m$ denote the set of images assigned to cluster m, and let $\boldsymbol{\mu}_m$ denote its centroid. For each cluster, one representative image is selected as the sample whose embedding lies closest to the centroid,

$$I_m^* = \arg\min_{I^{(i)} \in \mathcal{C}_m} d\Big(\mathbf{f}^{(i)}, \boldsymbol{\mu}_m\Big). \tag{4.1}$$

This yields k central representatives of the dominant normal modes. The remaining $N-k$ reference slots are then filled by boundary top-up. To this end, clusters are ordered by decreasing cluster radius,

$$r_m = \max_{I^{(i)} \in \mathcal{C}_m} d\Big(\mathbf{f}^{(i)}, \boldsymbol{\mu}_m\Big), \tag{4.2}$$

and additional samples are selected by iterating over these clusters and, at each pass, choosing the image farthest from the corresponding centroid until the N-sample budget is reached. The resulting sampled reference set $\mathcal{R}_s \subset \mathcal{P}_s$ is used as the SKU-specific reference memory for the downstream patch-level anomaly detector and combines central representatives of the dominant normal modes with additional support for less frequent but still valid normal appearances. The relative effectiveness of this strategy compared with alternative sampling approaches is evaluated in Section 5.3.

4.2.3. Reference Vocabulary Construction

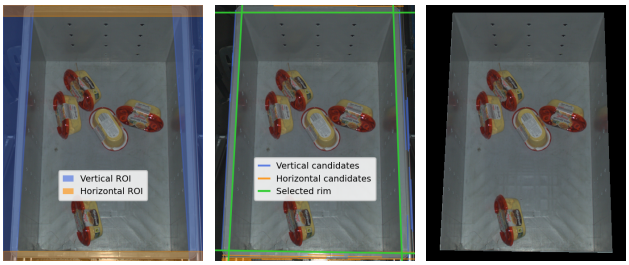
For each SKU s, a SKU-specific reference vocabulary $\mathcal{V}_s$ is constructed from the same sampled reference set $\mathcal{R}_s$ used by the anomaly detection branch. Text is extracted with EasyOCR [24] after central pre-cropping, using an OCR-specific resize setting. To improve robustness to product orientation, OCR outputs from four image rotations are cleaned, tokenised, and aggregated into $\mathcal{V}_s$. This vocabulary is later used by the variant verification branch for text-level matching against the expected SKU.

4.3. Anomaly Detection Branch

The anomaly detection branch, highlighted as the blue section in Figure 4.1, produces the raw anomaly map used by the downstream inference pipeline. It first applies container-interior masking to each query image I_q and then compares the resulting patch features with the SKU-specific reference memory bank $\mathcal{M}_s$. This yields the patch-level anomaly map $A(I_q)$ and image-level anomaly score $S(I_q)$, which are passed to the false-positive mitigation branch.

4.3.1. Image Preprocessing

Container Interior Localisation - Before anomaly detection, each query image I_q is preprocessed to suppress irrelevant background content and concentrate the extracted patch features on the stock-container interior. The localisation procedure consists of five steps: (i) a fixed central crop removes background regions around the container; (ii) the cropped image is resized to the anomaly detection input resolution; (iii) container-type-specific search regions and line-detection settings are selected to account for differences in geometry and appearance across chilled totes, ambient totes, and crates; (iv) Hough-line candidates are extracted within these regions after greyscale conversion, local contrast enhancement, smoothing, and edge extraction, allowing rim localisation to adapt to small variations in container position and orientation; and (v) the innermost valid horizontal and vertical lines are selected as the visible container boundaries. Figure 4.4 illustrates the search regions, detected rim candidates, and resulting masked image.



(a) (b) (c)

Figure 4.4: Example of the container-interior localisation and masking procedure used during query-image preprocessing. (a) Container-type-specific vertical and horizontal search regions restrict the Hough-line search to plausible rim locations after fixed cropping and resizing. (b) Candidate vertical and horizontal line detections are filtered to select the innermost valid container rim. (c) The selected rim defines the valid container interior, producing the masked image passed to anomaly-map generation.

Patch Masking - The detected container boundaries define the valid interior region, which is converted into a patch-level mask so that only patch features inside the container are used for anomaly detection. This suppresses background features outside the container. No Hough-based masking is applied to reference images. Instead, reference samples undergo the same fixed pre-cropping and resizing, and are additionally rotated. This preserves consistent scale between query and reference images while avoiding the exclusion of valid edge-region content caused by imperfect container-rim estimation on the reference images. If a container boundary is not detected, a fallback boundary estimate is used to define a usable interior region. The resulting preprocessing procedure is SKU-agnostic and provides the input to the anomaly map generation described in Subsection 4.3.2.

4.3.2. Anomaly Map Generation

Motivation - For cluttered stock-container inspection, anomaly detection is formulated as a reference-based patch-matching problem. This formulation is motivated by three properties of the setting: the inspected SKU assortment is large and continuously changing, defective samples are scarce and heterogeneous, and many relevant anomalies are spatially localised rather than globally distributed. Anomaly detection is therefore performed using AnomalyDINO [4]. The method represents normal appearance through a memory bank of patch-level DINOv2 features and scores a query image according to how well its local regions match this reference memory.

Patch-Based Memory Matching - For each SKU s, the sampled reference set $\mathcal{R}_s$ from Subsection 4.2.2 is encoded with DINOv2 as a grid of patch-level embeddings, which are aggregated into the SKU-specific reference memory bank $\mathcal{M}_s$. During inference, the preprocessed query image I_q is encoded in the same patch-feature space, and each query patch is compared with its nearest neighbour in $\mathcal{M}_s$. Figure 4.5 summarises the reference-based matching procedure. The nearest-neighbour distances obtained from patch-wise matching define the patch-level anomaly scores a_i, which are then arranged spatially to form the anomaly map $A(I_q)$ and aggregated to obtain the image-level anomaly score $S(I_q)$, as defined in Section 3.3. In the present pipeline, nearest-neighbour retrieval is implemented using a FAISS (Facebook AI Similarity Search) index to reduce the computational cost of matching against the reference memory bank. The aggregation used to compute $S(I_q)$ makes the final anomaly decision depend primarily on the worst-matching local regions. This is appropriate because many relevant anomalies in stock-container inspection are spatially localised, while others may extend across multiple visible products. The resulting anomaly map is used as the input to the false-positive mitigation branch described in Section 4.4.

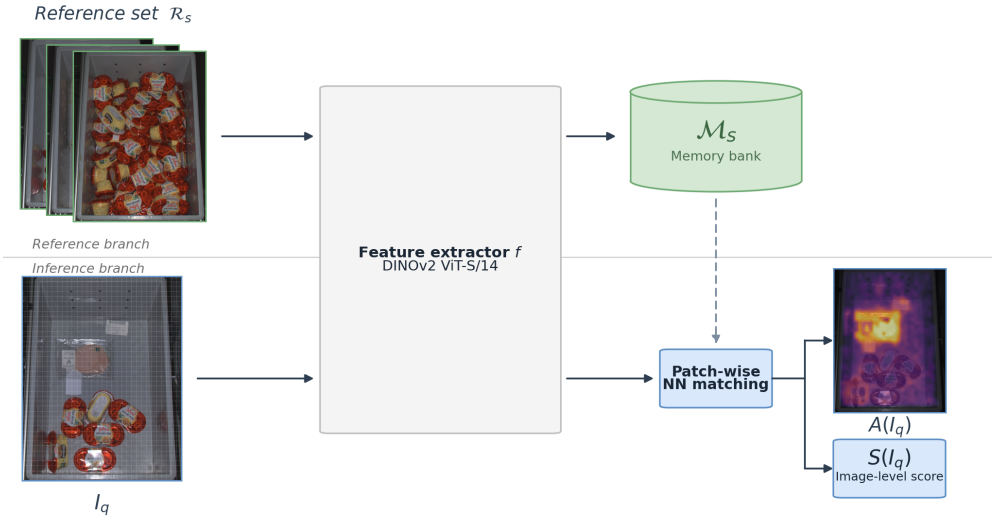


Figure 4.5: Reference-based patch-level anomaly detection for SKU-specific stock-container inspection. The sampled reference set $\mathcal{R}_s$ is encoded using the DINOv2 feature extractor f, and the resulting normal patch embeddings are stored in the reference memory bank $\mathcal{M}_s$. During inference, the query image I_q is encoded in the same feature space, after which each query patch is matched to its nearest neighbour in $\mathcal{M}_s$. The resulting patch-level distances are arranged spatially to obtain the anomaly map $A(I_q)$ and aggregated to compute the image-level anomaly score $S(I_q)$.

4.4. False-Positive Mitigation Branch

The false-positive mitigation branch, highlighted as the orange section in Figure 4.1, addresses a limitation of patch-based anomaly scoring in cluttered stock-container scenes. Although the reference-based anomaly detector identifies local deviations from the SKU-specific reference memory bank $\mathcal{M}_s$, it may also assign high scores to benign regions that differ from previously observed normal patterns. These false positives frequently arise from normal product-boundary transitions, container wear, loose transport material, and other nuisance regions that are not quality issues. To improve decision reliability, two complementary mitigation stages are introduced. First, product boundary extraction supports edge downweighting, which reduces anomaly responses near detected product boundaries where elevated scores often arise from normal context transitions rather than genuine defects. Second, candidate crop extraction isolates local high-response regions, after which debris filtering classifies these crops as benign debris or genuinely suspicious content, allowing nuisance patterns to be suppressed before the final image-level decision.

4.4.1. Edge Downweighting

Motivation - A frequent source of false positives in cluttered stock-container inspection arises at normal product boundaries. This is particularly common for products in transport packaging, trays, boxes, or other support structures, where the same product surface may appear against different surrounding context across otherwise normal observations. As a result, patch-based anomaly detection often produces elevated responses near product contours, even when the product itself is not defective. To reduce this effect, an edge downweighting step is applied to the anomaly map as post-processing.

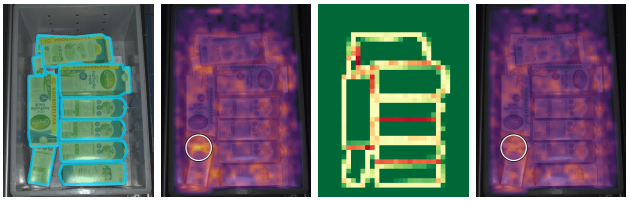
Product Boundary Extraction - Product instances are detected with YOLOv8s [25], producing bounding boxes that are used as prompts for MobileSAM [26] segmentation and subsequent mask refinement. The resulting product contours are used to identify boundary regions whose anomaly responses are treated as less reliable than responses from product interiors or the stock-container floor.

Edge-Based Reweighting - The product boundaries extracted from the refined instance masks are dilated inward and outward with a fixed radius to form an edge band. This edge band is projected onto the patch grid and used to construct an edge-based weight map $W(I_q) = (w_1, \ldots, w_P)$, where P denotes the number of query patches, $w_i \in [w_{\min}, 1]$, and $w_{\min} = 0.40$ is the configured minimum edge weight. Boundary patches are assigned lower weights, while non-boundary regions remain largely unchanged. The final edge-band geometry is selected empirically in Section 5.4.1, which evaluates the trade-off between false-positive suppression and recall cost across geometric configurations. The reweighted anomaly map is then given by

$$\tilde{A}(I_q) = A(I_q) \odot W(I_q), \tag{4.3}$$

where $A(I_q)$ denotes the original patch-level anomaly map, $\tilde{A}(I_q)$ the reweighted anomaly map, and $\odot$ element-wise multiplication. Figure 4.6 illustrates

this procedure for a boundary-driven false-positive example. The reweighted anomaly map is passed to the debris-filtering stage described in Subsection 4.4.2.



(a) (b) (c) (d)

Figure 4.6: Illustration of edge downweighting for a boundary-driven false positive. (a) Product instances are detected and segmented, and each product contour is dilated inward and outward to form an edge band. (b) The raw anomaly map $A(I_q)$ assigns an elevated response to a normal product-boundary region. The circle marks the boundary-driven hotspot responsible for the false-positive response. (c) The edge-based weight map $W(I_q)$ identifies boundary patches to be downweighted. (d) Applying $W(I_q)$ suppresses the boundary-driven response, yielding the reweighted anomaly map $\tilde{A}(I_q)$. The same circled region is suppressed after reweighting.

4.4.2. Debris Filtering

Motivation - While edge downweighting reduces false positives near product boundaries, cluttered stock-container scenes still contain benign nuisance regions that may produce strong anomaly responses, including debris, label remnants, transport material, and container-related wear. To suppress such nuisance patterns, a debris filtering stage is applied after edge downweighting. The stage consists of candidate crop extraction, lightweight rule-based pre-filtering, and crop-level classification. The complete debris-filtering sequence is summarised in Figure 4.7.

Candidate Crop Extraction - Suspicious local regions are extracted from the reweighted anomaly map $\tilde{A}(I_q)$. Crop extraction is gated by an image-level anomaly threshold, such that crops are generated only for sufficiently anomalous images. Within these images, the reweighted anomaly map is thresholded to identify high-response regions, connected components are extracted on the patch grid, and very small components are removed. The remaining components are mapped back to pixel space and cropped from the corresponding preprocessed query image, producing a compact set of suspicious local regions for further analysis.

Rule-Based Pre-Filtering - Before CNN-based crop classification, two lightweight rule-based filters are applied to suppress likely benign candidate regions. The *wall filter* removes crops with excessive overlap with predefined stock-container wall regions, where anomaly responses often arise from stickers, labels, or reflections rather than from genuine quality issues. The *reflection filter* removes crops dominated by bright, low-saturation pixels in HSV space, which are characteristic of specular highlights on container surfaces or packaging film. Both filters operate directly on the cropped image region and require no learned parameters. They reduce the number of crops forwarded to the debris classifier and remove nuisance regions that are more reliably identified by simple geometric and colour cues than by appearance-based classification.

Crop Classification and Decision Rule - Each extracted crop that remains after pre-filtering is classified as *debris* or *not debris* using a binary CNN classifier with a ResNet-18 backbone [27]. A CNN was chosen for this stage because debris filtering reduces to local visual classification on small cropped regions, where the relevant evidence is primarily carried by texture, material appearance, and local shape cues [28, 29]. ResNet-18 was selected as the final backbone based on the

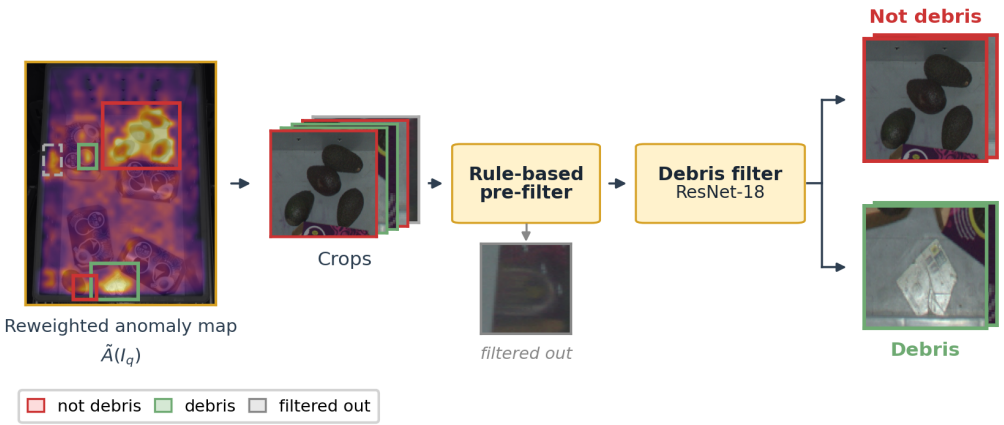


Figure 4.7: Overview of the debris-filtering stage. Candidate crops are extracted from the reweighted anomaly map $\tilde{A}(I_q)$ and processed by a rule-based pre-filter followed by a ResNet-18 debris classifier. Red boxes indicate suspicious *not debris* regions, green boxes indicate benign debris regions, and grey boxes indicate crops removed by the pre-filter.

classifier comparison reported in Subsection 5.4.2. Before inference, each crop is resized to 224×224 pixels, normalised, and forwarded through the classifier to obtain the probability that the region belongs to the *not debris* class. A threshold-based decision rule is then applied to control escalation conservatively. Let $\mathcal{C}(I_q)$ denote the set of crops that remain after pre-filtering for query image I_q, and let $\hat{y}_c \in \{0, 1\}$ denote the binary prediction for crop $c \in \mathcal{C}(I_q)$, where $\hat{y}_c = 1$ corresponds to *not debris*. Crop-level predictions are aggregated at image level using an OR rule,

$$\hat{y}_{\mathrm{df}}(I_q) = \bigvee_{c \in \mathcal{C}(I_q)} \hat{y}_c, \tag{4.4}$$

such that an image is considered anomalous if at least one crop is classified as *not debris*. This improves robustness against benign nuisance regions while preserving sensitivity to suspicious local content.

4.5. Variant Verification Branch

The variant verification branch, highlighted as the purple section in Figure 4.1, provides a text-level consistency check in parallel to visual anomaly detection. It compares OCR tokens extracted from the query image with the SKU-specific reference vocabulary $\mathcal{V}_s$ constructed in Subsection 4.2.3. The resulting decision complements the anomaly map by targeting wrong-variant cases that may be visually subtle in patch-feature space.

Motivation - Fine-grained product variants may differ primarily in printed packaging text, flavour names, volume indicators, or secondary labels, while retaining similar shape, colour, and layout. Such cases may not always produce a reliable anomaly score through reference-based patch matching alone. The variant verification branch therefore uses OCR-based vocabulary matching as an additional cue for detecting wrong product variants. Its effectiveness is limited to cases in which discriminative text is visible, legible, and captured by the OCR system. Visually similar variants without reliable textual evidence remain outside the scope of this branch.

Query-Time Decision Rule - At inference, the query image I_q undergoes central pre-cropping and multi-rotation OCR using the same OCR-specific settings as reference vocabulary construction. Detected word tokens are cleaned and compared with $\mathcal{V}_s$ using a conservative fuzzy-matching rule to remain robust to OCR noise. A token is treated as foreign only if it passes strict confidence and plausibility checks, does not match the reference vocabulary, and is not part of a small list of generic packaging terms that occur across many SKUs. Let $\mathcal{T}(I_q)$ denote the set of detected query tokens, and let $\alpha(t) \in \{0, 1\}$ indicate whether token $t \in \mathcal{T}(I_q)$ is mismatching with respect to $\mathcal{V}_s$. The image-level variant decision is then

$$\hat{y}_{\mathrm{vv}}(I_q) = \bigvee_{t \in \mathcal{T}(I_q)} \alpha(t), \tag{4.5}$$

such that I_q is flagged as a wrong variant if at least one mismatching token is detected. Because only high-confidence and plausibly valid foreign tokens contribute to this decision, the branch is designed to remain conservative in practice.

4.6. Computational Efficiency and Deployment Considerations

Motivation - The proposed pipeline including all branches is designed not only for anomaly detection performance, but also for compatibility with practical deployment constraints. In cluttered stock-container inspection, computational cost arises from multiple interacting stages, including DINOv2-based patch feature extraction, nearest-neighbour matching against the SKU-specific reference memory bank $\mathcal{M}_s$, product boundary extraction for edge downweighting, crop-level debris classification, and OCR-based variant verification. These stages do not contribute equally to end-to-end latency, because some operations are applied to every query image, some can run in parallel, and debris classification is activated only for images that pass the anomaly threshold. Computational efficiency must therefore be considered at the level of the full pipeline rather than for individual models in isolation.

Efficiency-Oriented Design Choices - Several design choices explicitly reduce runtime and memory usage while preserving detection robustness. A compact DINOv2-based backbone [6] is used to balance feature quality and inference speed, reference sampling limits memory-bank size, and pre-cropping together with container-interior masking reduce unnecessary background processing. For edge downweighting, YOLOv8s [25] and MobileSAM [26] are selected as lightweight detection and segmentation models that support product boundary extraction while providing a practical balance between accuracy and runtime. In addition, product boundary extraction for edge downweighting can be executed in parallel with the anomaly detection branch to reduce end-to-end latency. Debris filtering is applied conditionally, so crop extraction and crop-level classification are performed only when the initial anomaly score exceeds the detection threshold. These choices reflect a trade-off between robustness and efficiency, since larger reference sets and additional refinement stages can improve detection quality at the cost of higher runtime. The pipeline is therefore designed for practical deployability, balancing runtime efficiency with robustness rather than

minimising latency alone. Runtime behaviour and deployment feasibility are evaluated experimentally in Section 5.5.

5 Experiments

5.1. Experimental Setup

This section defines the common experimental protocol used to evaluate the proposed inspection pipeline and its components under controlled and comparable conditions.

Pipeline Configuration - Unless stated otherwise in the following experiments, all comparisons use a common default inspection configuration. This configuration fixes the shared preprocessing, reference sampling, anomaly detection, and false-positive mitigation settings. Images are preprocessed as described in Subsection 4.3.1. Reference sets are constructed using k-means clustering with boundary top-up and a default reference budget of 30 images per SKU. Anomaly detection is performed using the fixed AnomalyDINO-based detector described in Subsection 4.3.2. For false-positive mitigation, the default configuration applies symmetric edge downweighting with a radius of 12 pixels and a minimum edge weight of 0.4, followed by debris filtering with an image-level crop trigger threshold of 0.23, a patch-level crop threshold of 0.245, a maximum of seven candidate crops per image, and a debris-classifier threshold of 0.15.

This default configuration is used for the end-to-end evaluation and as the operating point in the component studies. Only the factor under investigation is varied in each experiment. The term *core detector* refers to the reference-based anomaly scoring pipeline before false-positive mitigation and variant verification. It includes image preprocessing, reference sampling, DINOv2 feature extraction, nearest-neighbour scoring, and image-level thresholding. The term *full pipeline* refers to the core detector augmented with edge downweighting, debris filtering, and variant verification.

Dataset - The main dataset was collected on the operational e-grocery line using fixed overhead RGB cameras mounted above the conveyor at multiple line locations and covers 20 SKU classes. For each SKU, the available images are split at container level into a candidate reference pool $\mathcal{P}_s$ and a held-out evaluation set, such that different images of the same physical container cannot appear in both partitions. The held-out evaluation set is not used for reference sampling or auxiliary model training. Thresholds and operating points are fixed before each reported test comparison, except where an experiment explicitly varies a threshold or states a separate operating-point selection rule.

The candidate reference pool is constructed from container images classified as normal by the reference-pool construction procedure described in Subsection 4.2.1, and contains only verified normal images. Across SKUs, the size of this pool varies substantially due to differences in product frequency, issue occurrence, and product size, ranging from approximately 250 to 1602 images and from 16 to 180 containers. From this pool, a fixed reference set $\mathcal{R}_s$ of 30 images per SKU is sampled for anomaly detection, corresponding to between 14 and 26 distinct containers depending on the SKU. The held-out evaluation set contains both normal and issue images, and all included images are manually reviewed to remove residual labelling errors from the earlier VLM-assisted filtering stage.

The 20 SKUs cover diverse product and packaging types, including dairy, produce, dry groceries, beverages, multipacks, and packaged convenience goods, and span five main issue families: wrong-SKU substitution, leakage, open packaging, multipack failures, and fresh-produce defects. The composition of the held-out evaluation set by issue family is summarised in Table 5.1, while detailed per-SKU counts and issue labels are provided in Appendix Table A.1.

Table 5.1: Composition of the held-out evaluation set by issue family.

Issue family	# SKUs	Good images	Issue images
Wrong-SKU*	4	599	344
Leakage	6	899	601
Open packaging	6	900	416
Multipack	2	298	195
Fresh produce	2	300	101
Overall*	20	2996	1657

* One wrong-SKU class is also reported separately in parts of the results because it constitutes an exceptional failure case involving an almost visually identical product variant with unchanged packaging and only a very subtle fine-grained product difference.

Evaluation Metrics - Performance is evaluated primarily at image level, since each image constitutes an individual observation processed by the anomaly detection pipeline. The main reported metrics are precision, recall, and F1-score, where TP, FP, TN, and FN denote the numbers of true positives, false positives, true negatives, and false negatives, respectively:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \qquad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}},$$

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

These metrics directly capture the trade-off between missed issues and unnecessary false alarms. Accuracy

is reported as a secondary summary metric:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}.$$

In addition, false-positive and false-negative counts are reported where relevant, since several component studies are intended to quantify how individual design choices shift the balance between nuisance alarms and missed anomalies. All reported image-level metrics are micro-averaged over the pooled evaluation set unless stated otherwise.

Implementation Details - The proposed pipeline is implemented in Python 3.10 using PyTorch 2.5. FAISS is used for nearest-neighbour retrieval in the reference memory bank, with GPU-based FAISS employed as the default configuration. Additional core libraries include OpenCV, NumPy, SciPy, scikit-learn, pandas, and ultralytics for product detection and segmentation.

The raw camera images have a resolution of 2464×2048 pixels. Before anomaly detection, a fixed percentage crop is applied to remove irrelevant border regions, after which the cropped container image is resized to an AnomalyDINO input resolution with smaller edge 448 pixels and centre-cropped to the nearest multiple of the DINO patch size, yielding 448×574 pixels in the default container setting. The object detector is evaluated at 640×640 pixels using YOLOv8s, while MobileSAM performs bounding-box-prompted segmentation on the pre-cropped image with internal long-side resizing to 1024 pixels. All debris-classifier crops are resized to 224×224 pixels.

Unless stated otherwise, all experiments use the same DINOv2 backbone (`dinov2_vits14`), the same FAISS retrieval configuration, the same YOLOv8s detector and MobileSAM segmentation setup, and the same ResNet-18 debris classifier. A fixed random seed is used for all stochastic sampling components. The complete inference configuration is reported in Appendix B. Training configurations for the YOLOv8s object detector and ResNet-18 debris classifier are reported in Appendix C. Runtime measurements are reported under batch-size-1 inference on an NVIDIA A100 GPU (80 GB). Warm-up iterations are excluded and CUDA synchronisation is applied before timing. Average latency in milliseconds per image is reported in the efficiency analysis in Section 5.5.

The subsequent sections evaluate end-to-end pipeline performance (Section 5.2), reference sampling strategies (Section 5.3), false-positive mitigation components (Section 5.4), computational efficiency (Section 5.5), and transferability to drone-based inspection (Section 5.6).

5.2. End-to-End Pipeline Evaluation

This section tests whether the proposed inspection pipeline improves operational decision reliability relative to the core detector, and identifies which individual components drive the resulting precision–recall trade-off. The full pipeline is first compared with the core detector as a whole. Incremental ablation then isolates the contribution of each stage in sequence.

Experiment: Full Pipeline vs. Core Detector - This experiment tests whether edge downweighting, debris filtering, and variant verification together improve operational inspection reliability relative to the core detector. Both configurations are evaluated on the full held-out evaluation set. The reference set, preprocessing, and anomaly detection configuration are held fixed, so that observed differences can be attributed solely to the presence or absence of these three additional decision stages.

Results: Full Pipeline vs. Core Detector - Table 5.2 shows that the full pipeline shifts the detector towards a higher-precision operating point relative to the core detector, increasing accuracy and F1-score while reducing recall. This shift reflects a clear trade-off: false positives drop from 1414 to 201 across the 2996 normal evaluation images, while false negatives rise from 177 to 646 across the 1657 issue images. Excluding one exceptional failure-case SKU, the full pipeline achieves 0.830 accuracy and 0.730 F1-score. The corresponding anomaly involves an almost visually identical product variant for which the packaging remains unchanged, while the only reliable cue is a very subtle fine-grained difference on the product itself.

Table 5.2: Overall image-level end-to-end performance of the core detector and the full pipeline.

Method	Acc.	Prec.	Rec.	F1
Core detector	0.658	0.511	0.893	0.650
Full pipeline	0.818	0.834	0.610	0.705
Core detector*	0.667	0.516	0.944	0.667
Full pipeline*	0.830	0.835	0.649	0.730

* Excluding one failure-case SKU.

Table 5.3 shows that performance varies considerably across issue families. The full pipeline performs best on wrong-SKU substitutions and multipack anomalies, which achieve the highest F1-scores. Leakage and open-packaging cases remain more challenging, as the anomaly evidence is often weaker, more localised, or visually similar to benign nuisance patterns. The lowest recall is observed for fresh-produce defects, indicating that subtle spoilage cues and viewpoint-dependent breakage remain difficult to distinguish reliably from

normal appearance variation. These results indicate that issue-family performance is not determined by anomaly severity alone, but also by how clearly the anomaly can be distinguished from normal packaging structure, boundary responses, and clutter. Detailed per-SKU performance results are provided in Appendix Table A.2.

Table 5.3: Overall image-level performance of the full pipeline grouped by issue family.

Issue family	# SKUs	Acc.	Prec.	Rec.	F1
Wrong-SKU*	3	0.928	0.904	0.889	0.897
Leakage	6	0.779	0.801	0.596	0.683
Open packaging	6	0.832	0.783	0.649	0.710
Multipack	2	0.856	0.913	0.703	0.794
Fresh produce	2	0.818	1.000	0.277	0.434

* Excluding one failure-case SKU.

Error Analysis - For the full pipeline, the remaining errors are concentrated in five recurring failure modes:

- **Reference-coverage false positives.** False positives occur when valid normal appearances are insufficiently represented in the reference set. Examples include transport packaging, unusual package orientations such as upside-down yoghurt tubs, and promotional markings.
- **Reflective and transparent packaging artefacts.** Condensation, surface reflections, and variable product content visible through transparent packaging can produce local anomaly responses that resemble genuine packaging defects.
- **Weak or boundary-adjacent anomaly evidence.** Small punctures, tears, minor leakage traces, and transparent leakage are difficult when the visual signal is weak, spatially limited, or close to product boundaries, where edge downweighting may suppress part of the relevant evidence.
- **Fresh-produce visibility limitations.** Fresh-produce failures are often view-dependent: breakage must be sufficiently exposed to the camera, while spoilage is typically small and partly obscured by plastic packaging.
- **Open packaging versus debris ambiguity.** Open packaging cases involving loose contents, such as rice, breadcrumbs, or salt, can be suppressed when the spilled product visually resembles benign dry debris rather than a clear packaging failure.

Experiment: Incremental Component Ablation - This ablation tests whether the performance shift from the raw AnomalyDINO detector to the full pipeline is distributed across the proposed pipeline stages or concentrated in a single component. Starting from the raw AnomalyDINO detector without the proposed additions, components are enabled step by step, each variant adding one stage to the previous configuration. Before the reference-sampling stage is added, the detector uses a fixed random reference subset of the same size. The evaluated sequence consists of preprocessing, reference sampling, edge downweighting, debris filtering, and variant verification. All variants use the same dataset split, reference pool, and decision protocol.

Results: Incremental Component Ablation - Table 5.4 shows that the operating-point shift is distributed across multiple stages. Debris filtering accounts for the largest reduction in false positives, while edge downweighting produces the largest single F1-score step. Preprocessing and reference sampling provide only modest overall changes relative to the baseline detector: preprocessing reduces false positives while reference sampling shifts the operating point marginally towards higher recall at the cost of a small increase in false positives, leaving overall F1 effectively unchanged. The first major shift occurs after edge downweighting, which reduces false positives by 280, but also lowers recall by suppressing some anomalies near product boundaries. The largest false-positive reduction is obtained after debris filtering, which removes a further 938 false positives at the cost of 440 additional false negatives and increases F1-score from 0.678 to 0.696, showing that residual nuisance detections are a dominant remaining error source in cluttered stock-container scenes. The final variant verification stage gives a smaller but targeted improvement, reducing false negatives by 21 and increasing F1-score from 0.696 to 0.705 by recovering a residual substitution failure case. The final operating point is therefore not caused by a single modification, but by a combination of stages that suppress different false-positive sources while introducing different recall costs. This trade-off is also visualised in Figure 5.1, where the dominant false-positive reduction occurs after edge downweighting and especially after debris filtering.

Table 5.4: Incremental ablation of the proposed inspection pipeline. Each row adds one component to the previous configuration.

Configuration	FP	FN	Acc.	Prec.	Rec.	F1
Baseline	1410	227	0.648	0.504	0.863	0.636
+ Preprocessing	1380	190	0.663	0.515	0.885	0.651
+ Ref. sampling	1414	177	0.658	0.511	0.893	0.650
+ Edge downw.	1134	227	0.708	0.558	0.863	0.678
+ Debris filter	196	667	0.815	0.835	0.597	0.696
+ Variant verif.	201	646	0.818	0.834	0.610	0.705

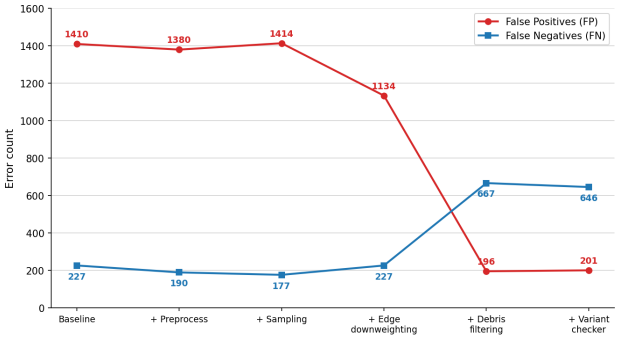


Figure 5.1: False positives and false negatives across the incremental ablation stages.

5.3. Reference Sampling Evaluation

This section evaluates how the construction of the SKU-specific reference set $\mathcal{R}_s$ affects the precision–recall behaviour of the proposed inspection pipeline. The evaluation first compares sampling strategies at a fixed budget and analyses whether the observed differences can be explained by geometric coverage and diversity in feature space. It then examines how detection performance changes as the reference budget varies.

5.3.1. Sampling Strategy Comparison

Experiment - This experiment tests whether the sampling strategy used to construct $\mathcal{R}_s$ affects the final inspection operating point. Specifically, it evaluates whether a reference set that balances representative coverage with limited support for rare normal appearances outperforms purely representative or purely diversity-driven selection. To isolate the effect of the sampling strategy, all methods use the same candidate reference pools, detector, preprocessing, evaluation set, and reference budget of 30 images per SKU. The evaluated methods include random sampling as an unstructured baseline, k-means clustering and facility-location sampling as representative selection methods, diversity-driven FPS sampling, and hybrid variants that combine representative coverage with additional support for rare normal appearances. Edge downweighting is applied throughout with outer and inner radii of 9 and 4 pixels. Although this differs slightly from the symmetric radius-12 setting used in the default final pipeline, it is kept fixed across all sampling methods and therefore does not affect the relative comparison.

For fair comparison between sampling methods, the image-level threshold was selected separately for each method using a fixed SKU-balanced operating rule. The selected threshold was the lowest value for which the average across SKUs of the ratio of FP_s to N_s remained at or below 0.30, with FP_s denoting the number of normal images of SKU s incorrectly classified as anomalous and N_s the total number of evaluated images for that SKU, including both normal and anomalous images. Under this rule, FPS used an image-level threshold of 0.245, k-means + FPS used 0.235, and all remaining methods used 0.230.

Results - Table 5.5 reports the resulting performance at both the core-detector stage and after false-positive mitigation under this threshold-selection rule.

Table 5.5: Comparison of reference sampling strategies at the core-detector stage and after false-positive mitigation. Methods are ordered by F1-score after false-positive mitigation.

	Core	After mitigation		
Method	**F1**	**Prec.**	**Rec.**	**F1**
k-means + boundary	0.6504	0.8253	0.6186	0.7071
Random	0.6514	0.7862	0.6415	0.7065
FPS	0.6481	0.8225	0.6126	0.7022
Facility-location	0.6531	0.8078	0.6204	0.7000
k-means + FPS	0.6581	0.8056	0.6174	0.6988
k-means	0.6527	0.7971	0.6210	0.6981
Facility-location + FPS	0.6463	0.8021	0.6082	0.6936

At the core-detector stage, the differences between methods remain moderate. The strongest raw detector is obtained with the k-means-plus-FPS hybrid. After edge downweighting and debris filtering, however, the ranking changes. The best result is obtained with k-means clustering plus boundary top-up, while random sampling and FPS remain competitive but slightly weaker. This shows that the most effective reference set is not the one that performs best at the raw detector stage, but the one that yields the best precision–recall trade-off after downstream false-positive mitigation.

The method comparison highlights three main findings. First, representative selection alone is not sufficient. Facility-location and pure k-means sampling remain competitive, but neither outperforms the best balanced method, indicating that common normal appearances must be complemented by limited support for less frequent but valid normal views. Second, diversity alone is also insufficient. FPS achieves high precision after false-positive mitigation, but this comes at the cost of lower recall, showing that broader reference support produces a precision–recall shift rather than a uniform improvement. Third, the way this additional support is introduced matters. Adding boundary samples to k-means clustering yields the best overall result, whereas adding FPS-based diversity to k-means sampling gives only a small improvement over pure k-means selection and remains weaker than the boundary-augmented variant. This indicates that structured support near the limits of

representative clusters is more effective than adding global diversity alone. Together, these results indicate that the strongest reference set is not obtained by maximising centrality or diversity alone, but by balancing representative coverage with a modest and structured extension towards less frequent normal variations.

A breakdown by issue family further clarifies the trade-off between the strongest methods. Figures 5.2 and 5.3 show that the main difference between k-means clustering plus boundary top-up and FPS arises for leakage and open-packaging cases. For leakage-related anomalies, the balanced k-means-based method yields fewer false negatives while maintaining a similar false-positive level, which explains much of its final advantage. By contrast, FPS performs better on a small number of open-packaging cases, indicating that broader variation in the reference set can help when the anomaly evidence is spatially more distinct.

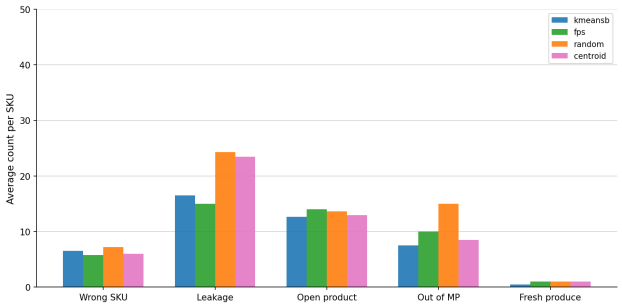


Figure 5.2: Average false positives per issue family for the main sampling methods.

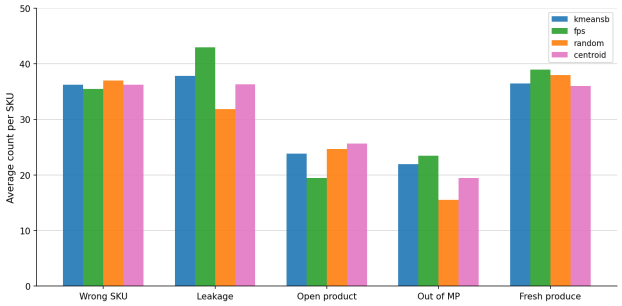


Figure 5.3: Average false negatives per issue family for the main sampling methods.

Coverage and Diversity Analysis - This analysis examines whether the precision–recall differences between sampling strategies can be explained by the geometric properties of the selected reference sets in feature space, specifically spread and mean nearest-neighbour distance (mNND). Here, mNND refers to the mean distance from the normal pool to the selected references. The results are interpreted by relating these properties to the precision and recall obtained after false-positive mitigation.

Table 5.6 shows two main patterns. First, recall appears to be strongly related to the spread of the selected reference set. Methods with lower spread ratio, most notably random sampling, retain the highest recall, whereas more strongly dispersed methods tend to miss more anomalies. This suggests that broader spread enlarges the region covered by the reference memory bank, so some anomalous patches obtain closer nearest-neighbour matches and are less likely to exceed the anomaly threshold.

Table 5.6: Coverage–diversity characteristics and performance after false-positive mitigation of the main sampling methods.

Method	mNND ↓	Spread	Prec.	Rec.	F1
Facility-loc.	0.0473	1.182	0.810	0.616	0.700
k-means	0.0480	1.180	0.797	0.621	0.698
Facility-loc. + FPS	0.0487	1.380	0.819	0.602	0.694
k-means + FPS	0.0500	1.445	0.821	0.608	0.699
k-means + boundary	0.0505	1.371	0.825	0.619	0.707
Random	0.0650	0.943	0.786	0.642	0.707
FPS	0.0702	1.528	0.823	0.613	0.702

Second, precision is not explained by average coverage alone. Facility-location and pure k-means sampling achieve the strongest average coverage of the normal pool, but do not obtain the highest precision. The best precision is instead achieved by k-means clustering with boundary top-up. This suggests that central coverage of the normal pool must be complemented by limited support for rare but valid normal appearances. Average coverage is therefore necessary but not sufficient; precision also depends on whether the reference set includes the less frequent normal appearances that would otherwise be flagged as anomalies.

Together, these results explain why k-means clustering with boundary top-up performs best overall. Random sampling attains a similar F1-score through higher recall, but at a larger false-positive cost, whereas FPS reduces false positives more effectively but loses recall as the reference set becomes too dispersed. The strongest overall result is therefore obtained by balancing central coverage with a modest and structured extension towards less frequent normal appearances.

5.3.2. Effect of Sampling Budget

Experiment - This experiment tests how the size of the SKU-specific reference set affects the final precision–recall trade-off and identifies the reference budget beyond which additional images provide diminishing practical returns. Increasing the budget can improve coverage of valid normal variation, but may also broaden the accepted normal region and suppress subtle anomaly evidence. The sampling strategy

is fixed to k-means clustering with boundary top-up, while the reference budget is varied over 10, 20, 30, 50, and 100 images per SKU. All other detector, preprocessing, mitigation, and evaluation settings are kept fixed.

Results - Figure 5.4 shows a clear precision–recall trade-off as the reference budget increases. Moving from 10 to 100 references steadily reduces the false-positive count from 395 to 147, but also increases the false-negative count from 483 to 837. As a result, larger reference sets produce higher precision but progressively lower recall. The highest F1-score is obtained at a reference budget of 10 images per SKU, where the detector operates with the tightest normal reference support, while larger budgets broaden the accepted normal region and suppress more anomaly evidence. At the same time, the substantial reduction in false positives between budgets of 10 and 30 images per SKU indicates that part of the added reference diversity remains useful for nuisance suppression. The budget of 30 images per SKU, used in the main strategy comparison, therefore represents a practical intermediate operating point rather than the F1-optimal one. In practice, this choice reduces false positives substantially relative to a budget of 10 images per SKU while avoiding the much stronger recall loss observed at budgets of 50 and 100 images per SKU.

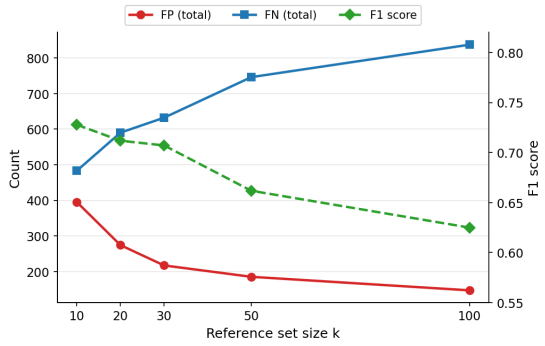


Figure 5.4: Effect of reference set size on total false positives, false negatives, and F1-score for k-means clustering with boundary top-up.

The budget sensitivity is not uniform across SKUs. The strongest recall losses with increasing reference set size occur for subtle leakage-related anomalies, whereas visually clearer substitution and packaging cases remain comparatively stable. This suggests that larger reference sets mainly hurt anomaly types whose evidence is already close to normal appearance variation, while offering little benefit for inherently hard-to-separate cases.

5.4. False-Positive Mitigation Evaluation

This section evaluates whether edge downweighting and debris filtering suppress the two main sources of clutter-induced false positives identified in the proposed pipeline: boundary-driven responses at normal product contours and nuisance responses from loose debris, labels, container wear, and transport material. The evaluation examines edge-band geometry, debris-filter stage composition, candidate-crop thresholds, and classifier backbone choice. The underlying anomaly detector, reference set, and evaluation data are held fixed throughout, so that observed changes can be attributed to the mitigation design choices under study.

5.4.1. Edge Downweighting

Experiment - This experiment tests whether edge downweighting reduces false positives caused by normal product contours, and how edge-band geometry affects the resulting false-positive versus false-negative trade-off. The detector, reference set, evaluation data, and downstream debris-filter configuration are held fixed, while only the edge-downweighting geometry is varied. The comparison includes a detector without downweighting, an outward-only band, a symmetric inward–outward sweep, and an asymmetric inward–outward sweep. These band designs span the main suppression alternatives between exterior-only masking and progressively stronger symmetric or asymmetric suppression around the detected product boundary. The same exceptional failure-case SKU excluded in the preceding end-to-end comparison is also excluded from this edge-geometry analysis.

Results - Figure 5.5 shows a clear trade-off across all tested configurations. Relative to the detector without edge downweighting, every downweighting variant reduces the false-positive count, but this is consistently accompanied by more false negatives. Stronger symmetric bands move furthest towards false-positive suppression, whereas outward-only and asymmetric designs preserve more anomaly evidence but reduce false positives less strongly. The main design question is therefore not whether edge downweighting changes the operating point, but how much boundary suppression can be introduced before the recall loss becomes disproportionate.

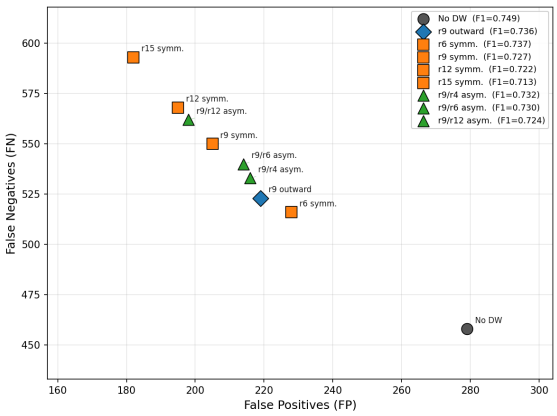


Figure 5.5: False-positive versus false-negative trade-off for the evaluated edge-downweighting configurations. All settings use the same detector, reference sampling, and post-processing pipeline, and differ only in edge-band geometry.

Figure 5.6 shows that the recall loss is concentrated in a small number of products. The clearest failure case is an avocado-in-sleeve product (SKU 11406433), for which the radius-12 symmetric band increases false negatives by 27 relative to no downweighting. For this SKU, the anomalous case consists of fruit located outside the sleeve, while part of the fruit is already visible in normal examples. Because this evidence lies directly in the region targeted by the suppression mask, downweighting weakens the anomaly signal disproportionately.

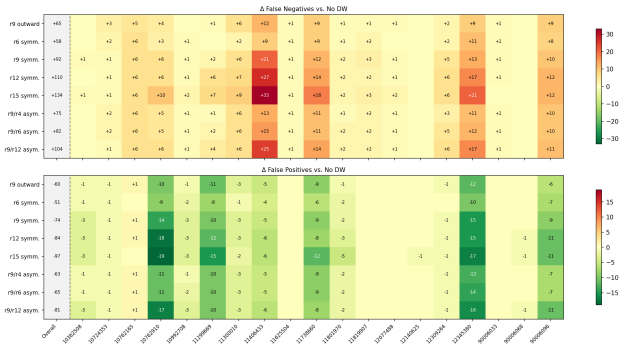


Figure 5.6: Per-SKU change in false negatives and false positives relative to the detector without edge downweighting. Positive values in the upper heatmap indicate increased false negatives, whereas negative values in the lower heatmap indicate reduced false positives. SKU identifiers correspond to the per-SKU dataset table in Table A.1.

A few additional leakage and open-packaging defects show a similar but less severe pattern, as their anomaly evidence is also boundary-adjacent, but here the false-negative increase is partly compensated by reduced false positives. By contrast, the strongest gains occur for products commonly transported in cardboard trays or similar transport units, whose constrained normal presentation leads to more boundary-driven nuisance responses in unusual stock-container configurations.

The selected operating point is the radius-12 symmetric band. Under this uniform setting, edge downweighting is applied to all evaluated SKUs and results in 84 fewer false positives at the cost of 110 additional false negatives relative to the detector without edge downweighting. This trade-off is selected because false positives are operationally costly in the downstream inspection process, whereas stronger suppression beyond this point yields only limited additional false-positive reduction at a steeper recall cost. This SKU-level failure mode indicates that future deployment configurations may benefit from SKU-conditional downweighting rules.

5.4.2. Debris Filtering

This subsection isolates which debris-filtering design choices are responsible for the false-positive reduction observed after edge downweighting. The underlying anomaly detector, reference set, evaluation data, and edge-downweighting configuration are held fixed throughout so that observed changes can be attributed to the evaluated debris-filter settings. Stage composition, candidate-crop extraction thresholds, and classifier backbone choice are evaluated in sequence.

Experiment: Stage Decomposition - An incremental ablation identifies which sub-components of the debris-filtering stage are responsible for the observed false-positive reduction and recall loss. The underlying detector, reference set, evaluation data, edge-downweighting configuration, and candidate-crop extraction settings are held fixed. Starting from the edge-downweighted pipeline, the crop classifier, wall filter, and reflection filter are enabled incrementally, so that each step isolates the contribution of one additional sub-component.

Results: Stage Decomposition - Table 5.7 shows that the crop classifier is responsible for almost the entire false-positive reduction of the stage, while also introducing most of its recall loss. By contrast, the wall and reflection filters act as targeted refinements. Their individual effect is small, but both remove a further subset of residual container-wall and brightness-driven false alarms.

Table 5.7: Stepwise contribution of the debris-filtering stage after edge downweighting. The detector, reference set, and candidate-crop extraction settings are kept fixed; only the debris classifier, wall filter, and reflection filter are enabled incrementally. Results exclude the same exceptional failure-case SKU as in the preceding edge-downweighting comparison.

Stage	FP	FN	ΔFP	ΔFN	P	R	F1
Edge downw.	1103	133	–	–	0.564	0.915	0.697
+ Debris clf.	219	557	-884	+424	0.820	0.642	0.721
+ Wall filter	207	563	-12	+6	0.828	0.638	0.721
+ Refl. filter	195	568	-12	+5	0.835	0.635	0.722

Experiment: Candidate-Crop Thresholds - The effect of candidate-crop extraction thresholds on the debris-filter operating point is examined by varying the image-level threshold and patch-level crop threshold across a grid of values. These thresholds determine which candidate crop regions are forwarded to the debris classifier. All remaining debris-filter settings are held fixed, including the crop classifier, wall filter, reflection filter, and debris-classifier threshold.

Results: Candidate-Crop Thresholds - Figure 5.7 shows that the patch threshold is the main driver of the debris-filter operating point, whereas the image threshold has only a secondary effect. Increasing the patch threshold steadily reduces false positives by forwarding fewer benign candidate regions to the classifier, but it also suppresses weak anomaly evidence and therefore increases false negatives. The selected operating point remains at an image threshold of 0.230 and a patch threshold of 0.245, which provides a practical balance within this trade-off. At lower patch thresholds, too many benign local responses are forwarded to the classifier, whereas higher thresholds increasingly suppress weak but valid anomalies, in particular low-contrast leakage cues.

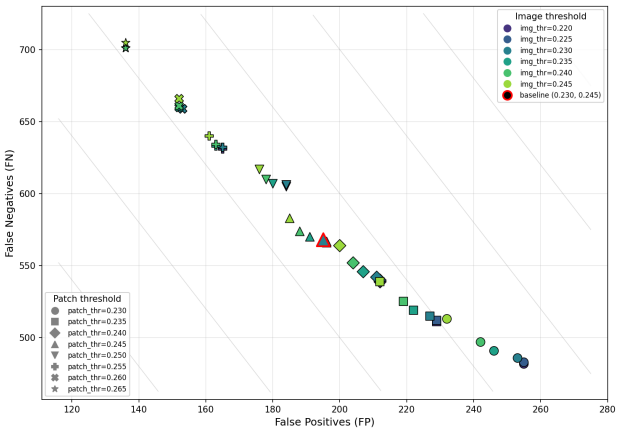


Figure 5.7: False-positive versus false-negative trade-off for the candidate-crop threshold sweep after edge downweighting. Colour indicates the image threshold and marker shape the patch threshold.

Experiment: CNN Backbone Selection - A comparison of four lightweight CNN classifiers, ResNet-18 [27], EfficientNet-B0 [30], MobileNet-V3-Small, and MobileNet-V3-Large [31], determines which backbone yields the most favourable false-positive versus false-negative operating curve for crop-level debris classification, and which offers the best balance between detection performance and inference efficiency. The candidate-crop extraction protocol and non-backbone debris-filter settings are kept fixed, and each backbone is evaluated over the same classifier-threshold sweep, so that the comparison isolates the effect of backbone choice.

Results: CNN Backbone Selection - Figure 5.8 shows that ResNet-18 forms the most favourable operating curve in the low-false-positive region of interest, with lower false-negative counts at the same false-positive level than the alternative backbones. MobileNet-V3-Large is the closest competitor, but it reaches its recall advantage only at higher false-positive levels. Table 5.8 shows that ResNet-18 is slower and larger than the MobileNet variants, yet the absolute latency remains limited since at most seven crops are evaluated per image. ResNet-18 is therefore selected as the final backbone.

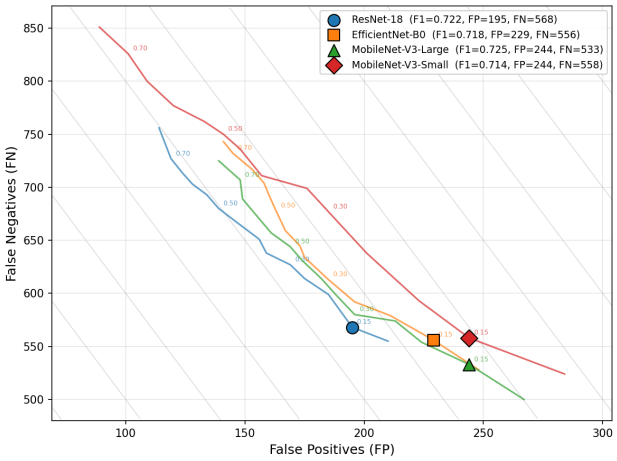


Figure 5.8: False-positive versus false-negative trade-off for the four evaluated debris-classifier backbones under a sweep of classifier thresholds. The highlighted markers indicate the selected operating point at a debris threshold of 0.15.

Table 5.8: Compact efficiency comparison of the evaluated debris-classifier backbones. Runtime is the mean per-crop latency on CPU (batch size 32).

Backbone	Params. [M]	Size [MB]	ms/crop
ResNet-18	11.18	44.8	36.7
EfficientNet-B0	4.01	16.3	59.2
MobileNet-V3-Large	4.20	17.0	17.5
MobileNet-V3-Small	1.52	6.2	6.97

5.5. Efficiency and Runtime Analysis

This section evaluates whether the proposed inspection pipeline satisfies the practical deployment constraint of approximately 10 s per image and identifies which components dominate end-to-end latency. The analysis focuses on total inference latency, component-wise runtime, and workload-dependent behaviour as scene complexity varies. The same hardware, implementation, reference set, detector configuration, and evaluation data are used throughout, so runtime differences reflect pipeline composition and input-dependent workload.

Experiment: End-to-End Latency - This experiment tests whether the proposed pipeline remains within the operational latency constraint when moving from the core detector to the full inspection system. The measured quantity is the total latency from image input to the final inspection decision, including anomaly detection, product boundary extraction, edge down-weighting, variant verification, and, where triggered, candidate crop extraction and debris filtering. Three pipeline variants are compared while the hardware, implementation, input resolution, detector configuration, and evaluation data are held fixed. Since several branches can execute in parallel, latency is analysed at system level rather than as the sum of individual component runtimes.

Results: End-to-End Latency - The results in Table 5.9 show that the proposed inspection system remains below the practical deployment constraint of approximately 10 s per image across all measured pipeline variants, with a full-pipeline 99th-percentile latency of 1,403 ms. The main latency increase occurs when the edge-downweighting variant is enabled, while the remaining stages add only a modest additional cost. This indicates that the dominant computational burden of the full pipeline arises from the additional product boundary extraction required for edge downweighting, rather than from the anomaly detector itself or from the later crop-based stages.

Table 5.9: End-to-end latency of the main pipeline variants in milliseconds under batch-size-1 inference on an NVIDIA A100 GPU. The symbol ‖ indicates parallel execution.

Pipeline variant	Mean	P50	P90	P99
Core detector	174	156	254	484
Core detector ‖ YOLO+SAM + edge downweighting	421	360	693	1,077
Full pipeline (parallel branches)	511	449	794	1,403

To better understand this behaviour, Table 5.10 reports the runtime of the main pipeline components separately. The component-wise timings show that the main increase from the core detector to the edge-downweighting variant is driven by the additional YOLO and MobileSAM product boundary extraction step, with MobileSAM remaining the highest-latency component under typical conditions. By contrast, edge downweighting itself adds only limited overhead once the masks have been generated. The remaining increase toward the full pipeline is most visible at the 99th percentile and arises primarily from the candidate crop extraction and debris-filter stages, which become more expensive on images with many suspicious regions requiring refinement. The variant verification stage does not determine this tail behaviour, since it executes in parallel and does not form the critical path under the measured operating conditions.

Table 5.10: Component-wise inference latency in milliseconds, measured from synchronised per-image timings.

Component	Mean	P50	P90	P99
AnomalyDINO	174	156	254	484
YOLO	119	98	190	421
MobileSAM	243	190	495	790
Edge downweighting	49	36	62	191
Candidate crop extraction	76	47	155	543
Debris filter	73	45	167	361
Variant verification	144	86	299	596

Experiment: Runtime Sensitivity - This experiment tests how input-dependent workload affects runtime, with emphasis on product-mask generation and crop-based debris filtering. The number of detected product instances is used as a proxy for segmentation workload, while the number of suspicious candidate crops determines the cost of the debris-filtering branch. The pipeline configuration, hardware, detector settings, and crop limit are held fixed, so that runtime variation can be attributed to scene complexity rather than configuration changes.

Results: Runtime Sensitivity - Figure 5.9 shows that MobileSAM latency scales approximately linearly with scene density, with a fitted slope of about 12 ms per additional instance. The median segmentation latency remains below 1 s across the observed range, rising from approximately 110 ms in sparse scenes to around 770 ms at 60 detected instances. By contrast, YOLO latency remains nearly constant across scenes and therefore contributes little to the observed runtime variation. The later crop-based refinement stages are likewise bounded in their effect on total latency. In particular, the debris-filter stage is capped at seven candidate crops per image and scales at approximately 28 ms per crop. High-density scenes are further concentrated in a small subset of SKUs, with three SKUs accounting for the majority of images containing more than 40 detected instances.

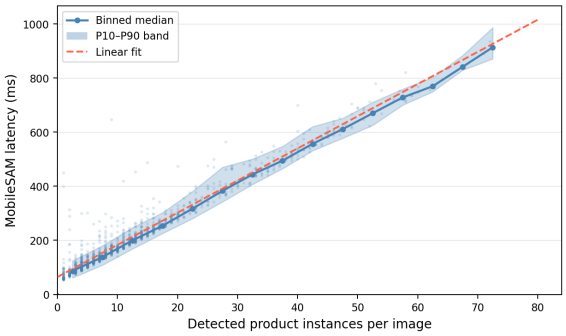


Figure 5.9: MobileSAM inference latency as a function of the number of detected product instances per image. Each point represents one image. The solid line shows the binned median, the shaded band indicates the P10–P90 range per bin, and the dashed line shows a linear fit with slope of approximately 12 ms per instance.

5.6. Transferability to Drone-Based Inspection

Mobile robotic inspection represents an operationally relevant extension of the proposed pipeline. In pallet-based inspection settings, products remain stationary rather than moving along a conveyor, so the imaging platform must move to the inspected scene instead of the products moving past a fixed camera. This section therefore evaluates whether the reference-based inspection principle remains applicable when images are acquired by a drone-mounted RGB camera. Rather than addressing full drone deployment, including navigation, safety, scheduling, or onboard processing, the evaluation isolates the core detector and assesses whether normal pallet observations can be separated from visible product deviations under mobile aerial acquisition. It further identifies which remaining pipeline components transfer directly, require recalibration, or need retraining on pallet-domain data.

Experiment - To isolate the effect of the acquisition domain, the fixed stock-container camera setup is replaced by a simulated drone-mounted RGB camera, while the core reference-based detector and image-wise inference protocol are retained. A pallet-inspection environment was constructed in Gazebo [32] with predefined pallet locations. The drone sequentially visits each pallet location, captures one overhead image, and continues to the next inspection pose. Figure 5.10 illustrates the simulated drone acquisition setup and representative rendered pallet scenes from which the top-down inspection images are obtained.

(a) (b) (c)

Figure 5.10: Qualitative examples from the drone-based pallet-inspection simulation. (a) Simulated drone acquisition above a predefined pallet location. (b) Rendered banana pallet scene containing normal bananas and visible ripeness deviations. (c) Rendered citrus pallet scene containing mandarins and orange wrong-SKU substitutions. The inspection pipeline is evaluated on top-down images derived from this simulated environment.

Two SKUs are evaluated: bananas and mandarins. For each SKU, a clean reference set $\mathcal{R}_s$ of 30 normal images is used. The test set contains 20 images per SKU, consisting of 10 normal and 10 issue images. Banana issues consist of five overripe and five underripe samples, whereas mandarin issues correspond to oranges as wrong-SKU substitutions. The transferred configuration reuses the core reference-based detector, consisting of a SKU-specific reference set, preprocessing, and patch-level anomaly scoring. The preprocessing settings are adapted to the drone flight height and expected pallet footprint. Edge downweighting and debris filtering are omitted because the simulated scenes represent idealised inspection conditions without realistic nuisance regions such as clutter, container wear, labels, transport material, or loose debris. Object detection and segmentation are evaluated only as transfer diagnostics.

Results - With the 30-image reference set per SKU, the normal and issue-image score distributions are perfectly separated. The highest normal score is 0.124, while the lowest issue score is 0.276. Any global threshold within this interval would therefore yield zero false positives and zero false negatives across the 40 simulated test images. Under these controlled conditions, bananas and mandarins both achieve precision, recall, and F1-score of 1.00. This result indicates that reference-based anomaly scoring can transfer effectively in the controlled simulated pallet setting when the reference set captures sufficient normal appearance variation. The result should not be interpreted as deployment validation, since the simulation assumes stable illumination, constant flight height, accurate camera positioning above each pallet, and no realistic pallet-domain debris.

Transfer Analysis - The most transferable component is the reference-based anomaly-scoring principle. In both the stock-container and pallet settings, the expected SKU is known, normality can be represented by a clean SKU-specific reference set, and visible issues are detected as local deviations from the reference memory bank. The main requirement is that the reference set must cover the normal appearance variation induced by the acquisition geometry.

Preprocessing transfers conceptually, but its parameters must be recalibrated for drone acquisition. Unlike the fixed conveyor setup, the pallet crop depends on flight height, camera intrinsics, pallet footprint, and drone positioning accuracy. The current setup assumes a visible pallet edge for Hough-line-based masking and approximate positioning above the pallet, which can accommodate small changes in position and rotation. Larger changes in height, viewing angle, or pallet placement would require additional image alignment and potentially reference data collected across the expected range of drone acquisition conditions.

Object detection is more domain-dependent than the reference-based scoring branch. In the diagnostic tests, the object detector did not transfer reliably: bananas were often detected as individual bananas rather than complete bunches, and oranges were sometimes detected as loose instances rather than grouped products. Mandarins were detected more reliably, likely because their appearance was closer to examples seen during detector training.

This indicates that the object detector should be retrained on pallet-domain products before being used for downstream mitigation. The segmentation model is more transferable because it is not trained for a specific product domain, but its practical performance remains dependent on the quality of the input bounding boxes.

Edge downweighting and debris filtering are expected to be less critical in real pallet scenes than in cluttered stock containers. Pallet layouts typically contain more uniform product orientation, less overlap, and fewer nuisance regions from stickers, container wear, and loose debris. Nevertheless, real pallet inspection may introduce different nuisance factors, including pallet wood, wrapping material, shadows, labels, and drone-induced image artefacts. These stages therefore require validation or retraining on representative pallet-domain imagery. Overall, the drone experiment supports the transferability of the reference-based anomaly scoring principle, while showing that preprocessing, object localisation, threshold calibration, and false-positive mitigation remain domain-specific.

6 Discussion

This chapter interprets the experimental findings in terms of the central design objective: detecting visible product anomalies in cluttered, multi-instance e-grocery stock containers while limiting false positives under warehouse deployment constraints. The experiments show that robust inspection in this setting cannot be achieved by maximising raw anomaly sensitivity alone. Normal product boundaries, packaging clutter, reflective surfaces, container wear, and loose debris can generate high patch-level responses that, without downstream filtering, produce a high volume of clutter-induced nuisance alarms. The proposed pipeline should therefore be interpreted as a modular inspection pipeline calibrated towards a deployment-oriented operating point, rather than as a uniformly stronger anomaly detector across all objectives. The following sections discuss the design implications of this finding, the limitations of the evaluated approach, and the transferability of the core scoring branch to related inspection domains.

Design Implications for Cluttered E-grocery Inspection - The central design implication is that false-positive mitigation must be treated as a core system requirement rather than as auxiliary post-processing. In cluttered stock containers, nuisance alarms carry direct operational consequences: each unnecessary alarm risks triggering manual intervention, interrupting downstream handling, and reducing trust in the inspection system. The precision–recall trade-off observed in the end-to-end evaluation should therefore not be interpreted as a limitation of the chosen anomaly detector, but as a structural consequence of operating in a setting where many high-scoring local responses are benign.

Edge downweighting and debris filtering address this requirement through complementary mechanisms. Edge downweighting suppresses predictable boundary-driven responses at normal product contours, but can also weaken anomaly evidence that lies near the same boundaries. Debris filtering validates local high-scoring regions and removes benign nuisance responses, but can suppress true anomalies when open packaging or leakage evidence resembles benign debris. Reference sampling supports the same objective by improving coverage of valid normal appearances, whereas variant verification shows that narrow auxiliary modules can recover specific false negatives that patch-level matching alone does not capture. Together, these stages address distinct failure modes in sequence, each shifting the precision–recall operating point in a way that must be balanced against the operational cost of false alarms and missed anomalies.

This requirement follows from the way patch-level anomaly evidence is converted into image-level inspection decisions. Although patch-level nearest-neighbour scoring is well suited to this setting because visible product and packaging anomalies are typically localised, the raw image-level anomaly score is derived from the highest-scoring patch responses across the full image. The experiments show that, in cluttered stock containers, such high-scoring regions are not necessarily operationally relevant: benign nuisance regions can trigger a false alarm, while suppressing those regions can also remove evidence for genuine anomalies. Anomaly maps should therefore be treated as candidate evidence rather than final decisions. For deployment, local anomaly responses require location-based and appearance-based validation before they determine the final accept-or-reject outcome.

Limitations, Failure Modes, and Parameter Sensitivity - The main source of remaining false positives is incomplete coverage of valid normal appearance variation. Even with SKU-specific reference sets and downstream mitigation stages, uncommon orientations, transport packaging, promotional markings, reflective surfaces, and transparent packaging can remain underrepresented. These appearances are operationally normal, but can produce feature-space deviations from the selected reference memory bank. The automatic reference construction process can reinforce this limitation when visually unusual but valid normal images are rejected during filtering. The reliability of automatic filtering depends on template image quality, product metadata, prompt design, and the ability of the vision-language model to distinguish true issue images from rare normal presentations.

The main source of remaining false negatives is limited or ambiguous anomaly evidence in a single overhead RGB image. Small punctures, minor tears, and low-contrast leakage traces may provide too little spatial support to dominate the anomaly map or to be selected reliably during crop extraction. When such cues lie near a product boundary, edge downweighting can suppress the available signal further. Fresh-produce defects introduce an additional visibility constraint: physical breakage must be exposed to the camera, while spoilage is often observed through plastic packaging where reflections, condensation, and colour variation reduce the available discriminative signal. Open packaging failures introduce a different ambiguity: loose contents such as breadcrumbs or salt can closely resemble benign dry debris at crop level. In these cases, the debris filter may suppress true-positive crops that the core detector correctly identified. These examples show that the proposed pipeline is limited not only by the detector and mitigation stages, but

also by the discriminative evidence available in overhead RGB imagery.

The reported operating point is also conditional on several interacting parameter choices, including the reference budget, sampling strategy, image-level threshold, edge-band geometry, crop extraction thresholds, and debris-classifier threshold. These settings were selected through systematic empirical evaluation, but they should not be interpreted as universally optimal across all product groups. In particular, SKUs with highly reflective or transparent packaging, products with large normal appearance variation, and sleeve-type produce with boundary-adjacent anomaly evidence may require different operating points or product-group-specific rules. Practical deployment therefore requires monitoring, recalibration, and periodic reference-set maintenance as packaging, promotions, and imaging conditions change.

Transferability and Deployment Scope - The transfer experiment indicates that the reference-based scoring principle is not tied to the fixed-conveyor acquisition setup. Under controlled drone-simulation conditions, SKU-specific normality remains representable through a compact reference set, and visible deviations remain detectable as local patch-feature mismatches. However, this transfer applies primarily to the core scoring branch, reflecting the training-free, reference-based design inherited from AnomalyDINO [4]. Preprocessing, object localisation, threshold selection, and false-positive mitigation all depend on acquisition geometry and scene-specific nuisance factors, and require recalibration or retraining when applied to a new inspection domain. This supports a modular deployment view: the reference-based scoring branch is less domain-specific than the surrounding calibration and false-positive mitigation stages, which must be adapted to the acquisition context. The drone experiment should therefore be interpreted as evidence of conceptual transferability rather than as validation of a drone-based deployment system.

Implications for Reference-Based Anomaly Detection - Taken together, the results position the proposed pipeline as a deployment-oriented extension of reference-based anomaly detection for cluttered stock-container inspection. PatchCore and AnomalyDINO show that patch-level memory matching with strong visual features provides an effective basis for data-efficient industrial anomaly detection [3, 4]. However, recent evidence from the Kaputt benchmark shows that standard anomaly detection methods struggle under the substantial pose, appearance, and reference variability typical of product inspection settings [21]. The present results support the same conclusion in the cluttered, multi-instance stock-container setting: strong patch-level features are necessary, but not sufficient for operational reliability.

7 Conclusion

Summary - This thesis addressed how a reference-based anomaly detection pipeline can be designed to detect visible anomalies in cluttered, multi-instance e-grocery stock containers while limiting false positives and remaining scalable under warehouse deployment constraints. The results show that this can be achieved by automating reference-set construction and treating false-positive mitigation as a core system requirement rather than as auxiliary post-processing. The resulting design combines diversity-aware SKU-specific reference sets, patch-level nearest-neighbour matching of DINOv2 features for localised deviation detection, edge downweighting and debris filtering for clutter suppression, and OCR-based variant verification for visually subtle wrong-SKU cases. Together, these components define a modular inspection pipeline that achieves a deployment-oriented precision–recall trade-off while satisfying practical latency and scalability constraints.

Findings - The end-to-end evaluation shows that the full pipeline substantially reduces false positives relative to the core detector, shifting the operating point towards higher precision at a recall cost. This recall cost is not evenly distributed across anomaly types: wrong-SKU substitutions and multipack failures are detected most reliably, while leakage, open packaging, and fresh-produce defects remain challenging due to weaker or more visually ambiguous anomaly evidence. The false-positive reduction is driven mainly by edge downweighting and debris filtering, which target structurally different nuisance sources, while variant verification recovers a narrow but operationally relevant subset of visually subtle wrong-SKU cases.

Among the evaluated reference sampling strategies, k-means clustering with boundary top-up yields the most favourable final operating point by balancing coverage of common normal appearances with representation of less frequent but valid normal views. The runtime analysis further shows that the full pipeline operates well within the operational latency constraint, with a 99th-percentile latency of 1.4 s against the 10 s warehouse latency budget.

Future Work - Future work should first extend the evaluation beyond the twenty-SKU scope of the present study. A larger-scale evaluation would test whether the observed precision–recall trade-off remains stable across broader SKU diversity, packaging

types, and normal appearance variation. Such an extension would also provide additional false-positive and false-negative crops for retraining the debris filter on a wider range of nuisance and anomaly appearances. This evaluation could also assess whether product-group-specific thresholds or component settings improve the operating point without introducing excessive calibration overhead.

A further direction concerns the maintenance of SKU-specific reference pools and reference sets beyond initial construction. Consistently low reference-pool acceptance rates or recurring false positives should flag SKUs for review, since these patterns may indicate missing normal appearances or overly restrictive automatic reference-pool filtering. For such SKUs, the VLM labelling prompts, SKU metadata, and template imagery used during automatic construction could be updated, including additional template views for highly pose-dependent products. Packaging redesigns or changes in promotional markings should trigger reference-pool reconstruction and subsequent resampling of the reference set, as the existing references may no longer model current normality.

Finally, future work should address anomaly types for which the limiting factor is the available evidence in overhead RGB imagery. Complementary imaging modalities within the existing acquisition setup, such as near-infrared or hyperspectral imaging, could be evaluated for selected product groups where colour and texture cues are weak or obscured by packaging [33]. Fresh-produce spoilage and transparent leakage are the most direct applications, as both may produce limited discriminative signal in RGB imagery alone.

Concluding Remarks - This thesis establishes systematic false-positive mitigation as a necessary complement to reference-based anomaly detection in cluttered, multi-instance e-grocery stock-container inspection. In the broader fulfilment setting, automated inspection is valuable only if it improves product-quality control without creating a manual-review burden that outweighs its operational benefit. The proposed pipeline defines a practical operating regime in which the expected SKU is known, representative normal references are available, and visible product or packaging deviations must be detected with limited tolerance for nuisance alarms. The results indicate that robust inspection in this setting depends not only on the core anomaly detector, but also on the construction and maintenance of clean reference sets and the targeted suppression of clutter-induced nuisance responses.

References

[1] Paul Bergmann et al. 'MVTec AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection'. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA: IEEE, June 2019, pp. 9584–9592. ISBN: 978-1-7281-3293-8. DOI: 10.1109/CVPR.2019.00982. URL: https://ieeexplore.ieee.org/document/8954181/ (visited on 22/10/2025).

[2] Yang Zou et al. *SPot-the-Difference Self-Supervised Pre-training for Anomaly Detection and Segmentation*. July 2022. DOI: 10.48550/arXiv.2207.14315. URL: http://arxiv.org/abs/2207.14315 (visited on 22/10/2025).

[3] Karsten Roth et al. 'Towards Total Recall in Industrial Anomaly Detection'. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. New Orleans, LA, USA: IEEE, June 2022, pp. 14298–14308. ISBN: 978-1-6654-6946-3. DOI: 10.1109/CVPR52688.2022.01392. URL: https://ieeexplore.ieee.org/document/9879738/ (visited on 14/10/2025).

[4] Simon Damm et al. *AnomalyDINO: Boosting Patch-based Few-shot Anomaly Detection with DINOv2*. Mar. 2025. DOI: 10.48550/arXiv.2405.14529. URL: http://arxiv.org/abs/2405.14529 (visited on 14/10/2025).

[5] Zhaopeng Gu et al. 'AnomalyGPT: Detecting Industrial Anomalies Using Large Vision-Language Models'. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 38.3 (Mar. 2024), pp. 1932–1940. ISSN: 2374-3468, 2159-5399. DOI: 10.1609/aaai.v38i3.27963. URL: https://ojs.aaai.org/index.php/AAAI/article/view/27963 (visited on 14/10/2025).

[6] Maxime Oquab et al. *DINOv2: Learning Robust Visual Features without Supervision*. Feb. 2024. DOI: 10.48550/arXiv.2304.07193. URL: http://arxiv.org/abs/2304.07193 (visited on 22/10/2025).

[7] Paul Bergmann et al. 'Uninformed Students: Student-Teacher Anomaly Detection With Discriminative Latent Embeddings'. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, June 2020, pp. 4182–4191. ISBN: 978-1-7281-7168-5. DOI: 10.1109/CVPR42600.2020.00424. URL: https://ieeexplore.ieee.org/document/9157778/ (visited on 14/10/2025).

[8] Vitjan Zavrtanik et al. 'DRÆM – A discriminatively trained reconstruction embedding for surface anomaly detection'. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal, QC, Canada: IEEE, Oct. 2021, pp. 8310–8319. ISBN: 978-1-6654-2812-5. DOI: 10.1109/ICCV48922.2021.00822. URL: https://ieeexplore.ieee.org/document/9710329/ (visited on 14/10/2025).

[9] Denis Gudovskiy et al. 'CFLOW-AD: Real-Time Unsupervised Anomaly Detection with Localization via Conditional Normalizing Flows'. In: *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Waikoloa, HI, USA: IEEE, Jan. 2022, pp. 1819–1828. ISBN: 978-1-6654-0915-5. DOI: 10.1109/WACV51458.2022.00188. URL: https://ieeexplore.ieee.org/document/9707081/ (visited on 14/10/2025).

[10] Julian Wyatt et al. 'AnoDDPM: Anomaly Detection with Denoising Diffusion Probabilistic Models using Simplex Noise'. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. New Orleans, LA, USA: IEEE, June 2022, pp. 649–655. ISBN: 978-1-6654-8739-9. DOI: 10.1109/CVPRW56347.2022.00080. URL: https://ieeexplore.ieee.org/document/9857019/ (visited on 14/10/2025).

[11] Jongheon Jeong et al. 'WinCLIP: Zero-/Few-Shot Anomaly Classification and Segmentation'. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Vancouver, BC, Canada: IEEE, June 2023, pp. 19606–19616. ISBN: 979-8-3503-0129-8. DOI: 10.1109/CVPR52729.2023.01878. URL: https://ieeexplore.ieee.org/document/10204096/ (visited on 20/10/2025).

[12] Alec Radford et al. 'Learning Transferable Visual Models From Natural Language Supervision'. In: *Proceedings of the 38th International Conference on Machine Learning*. PMLR, July 2021, pp. 8748–8763. URL: https://proceedings.mlr.press/v139/radford21a.html (visited on 28/10/2025).

[13] Liron Bergman et al. *Deep Nearest Neighbor Anomaly Detection*. Feb. 2020. DOI: 10.48550/arXiv.2002.10445. URL: http://arxiv.org/abs/2002.10445 (visited on 14/10/2025).

[14] Marco Rudolph et al. 'Same Same But DifferNet: Semi-Supervised Defect Detection with Normalizing Flows'. In: *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*. Waikoloa, HI, USA: IEEE, Jan. 2021, pp. 1906–1915. ISBN: 978-1-6654-0477-8. DOI: 10.1109/WACV48630.2021.00195. URL: https://ieeexplore.ieee.org/document/9423203/ (visited on 14/10/2025).

[15] Xinyi Zhang et al. 'Unsupervised Surface Anomaly Detection with Diffusion Probabilistic Model'. In: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. Paris, France: IEEE, Oct. 2023, pp. 6759–6768. ISBN: 979-8-3503-0718-4. DOI: 10.1109/ICCV51070.2023.00624. URL: https://ieeexplore.ieee.org/document/10377534/ (visited on 14/10/2025).

[16] Qihang Zhou et al. *AnomalyCLIP: Object-agnostic Prompt Learning for Zero-shot Anomaly Detection*. Apr. 2025. DOI: 10.48550/arXiv.2310.18961. URL: http://arxiv.org/abs/2310.18961 (visited on 20/10/2025).

[17] Kilian Batzner et al. 'EfficientAD: Accurate Visual Anomaly Detection at Millisecond-Level Latencies'. In: *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. Waikoloa, HI, USA: IEEE, Jan. 2024, pp. 127–137. ISBN: 979-8-3503-1892-0. DOI: 10.1109/WACV57701.2024.00020. URL: https://ieeexplore.ieee.org/document/10484326/ (visited on 14/10/2025).

[18] Niv Cohen and Yedid Hoshen. *Sub-Image Anomaly Detection with Deep Pyramid Correspondences*. Feb. 2021. DOI: 10.48550/arXiv.2005.02357. URL: http://arxiv.org/abs/2005.02357 (visited on 14/10/2025).

[19] Zhaopeng Gu et al. 'UniVAD: A Training-free Unified Model for Few-shot Visual Anomaly Detection'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2025, pp. 15194–15203. URL: https://openaccess.thecvf.com/content/CVPR2025/html/Gu_UniVAD_A_Training-free_Unified_Model_for_Few-shot_Visual_Anomaly_Detection_CVPR_2025_paper.html (visited on 28/10/2025).

[20] Paul Bergmann et al. 'Beyond Dents and Scratches: Logical Constraints in Unsupervised Anomaly Detection and Localization'. In: *International Journal of Computer Vision* 130.4 (Apr. 2022), pp. 947–969. ISSN: 0920-5691, 1573-1405. DOI: 10.1007/s11263-022-01578-9. URL: https://link.springer.com/10.1007/s11263-022-01578-9 (visited on 24/10/2025).

[21] Sebastian Höfer et al. *Kaputt: A Large-Scale Dataset for Visual Defect Detection*. Oct. 2025. DOI: 10.48550/arXiv.2510.05903. URL: http:

//arxiv.org/abs/2510.05903 (visited on 24/10/2025).

[22] Patrick Follmann et al. 'MVTec D2S: Densely Segmented Supermarket Dataset'. In: *Computer Vision – ECCV 2018*. Ed. by Vittorio Ferrari et al. Vol. 11214. Cham: Springer International Publishing, 2018, pp. 581–597. ISBN: 978-3-030-01249-6. DOI: 10.1007/978-3-030-01249-6_35. URL: https://link.springer.com/10.1007/978-3-030-01249-6_35 (visited on 24/10/2025).

[23] Chaitanya Mitash et al. *ARMBench: An Object-centric Benchmark Dataset for Robotic Manipulation*. Mar. 2023. DOI: 10.48550/arXiv.2303.16382. URL: http://arxiv.org/abs/2303.16382 (visited on 24/10/2025).

[24] Jaided AI. *EasyOCR*. Sept. 2024. URL: https://github.com/JaidedAI/EasyOCR (visited on 29/04/2026).

[25] Glenn Jocher et al. *Ultralytics YOLO*. Jan. 2023. URL: https://github.com/ultralytics/ultralytics (visited on 28/04/2026).

[26] Chaoning Zhang et al. *Faster Segment Anything: Towards Lightweight SAM for Mobile Applications*. July 2023. DOI: 10.48550/arXiv.2306.14289. URL: http://arxiv.org/abs/2306.14289 (visited on 28/04/2026).

[27] Kaiming He et al. 'Deep Residual Learning for Image Recognition'. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, June 2016, pp. 770–778. ISBN: 978-1-4673-8851-1. DOI: 10.1109/CVPR.2016.90. URL: http://ieeexplore.ieee.org/document/7780459/ (visited on 20/04/2026).

[28] Wieland Brendel and Matthias Bethge. *Approximating CNNs with Bag-of-local-Features models works surprisingly well on ImageNet*. Mar. 2019. DOI: 10.48550/arXiv.1904.00760. URL: http://arxiv.org/abs/1904.00760 (visited on 20/04/2026).

[29] Sean Bell et al. 'Material recognition in the wild with the Materials in Context Database'. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston, MA, USA: IEEE, June 2015, pp. 3479–3487. ISBN: 978-1-4673-6964-0. DOI: 10.1109/CVPR.2015.7298970. URL: http://ieeexplore.ieee.org/document/7298970/ (visited on 20/04/2026).

[30] Mingxing Tan and Quoc Le. 'EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks'. In: *Proceedings of the 36th International Conference on Machine Learning*. PMLR, May 2019, pp. 6105–6114. URL: https://proceedings.mlr.press/v97/tan19a.html (visited on 29/04/2026).

[31] Andrew Howard et al. *Searching for MobileNetV3*. Nov. 2019. DOI: 10.48550/arXiv.1905.02244. URL: http://arxiv.org/abs/1905.02244 (visited on 29/04/2026).

[32] N. Koenig and A. Howard. 'Design and use paradigms for Gazebo, an open-source multi-robot simulator'. In: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Vol. 3. Sept. 2004, 2149–2154 vol.3. DOI: 10.1109/IROS.2004.1389727. URL: https://ieeexplore.ieee.org/document/1389727 (visited on 06/05/2026).

[33] Guoling Wan et al. 'Hyperspectral imaging technology for nondestructive identification of quality deterioration in fruits and vegetables: a review'. In: *Critical Reviews in Food Science and Nutrition* 65.32 (Dec. 2025), pp. 7923–7952. ISSN: 1040-8398. DOI: 10.1080/10408398.2025.2487134. URL: https://doi.org/10.1080/10408398.2025.2487134 (visited on 10/05/2026).

[34] Jinwon An and Sungzoon Cho. *Variational Autoencoder Based Anomaly Detection Using Reconstruction Probability*. Technical Report SNUDM-TR-2015-03. Seoul National University, 2015, pp. 1–18. URL: https://dm.snu.ac.kr/static/docs/TR/SNUDM-TR-2015-03.pdf (visited on 28/10/2025).

[35] Bo Zong et al. 'Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection'. In: *International Conference on Learning Representations*. Feb. 2018. (Visited on 28/10/2025).

[36] Samet Akcay et al. *GANomaly: Semi-Supervised Anomaly Detection via Adversarial Training*. Nov. 2018. DOI: 10.48550/arXiv.1805.06725. arXiv: 1805.06725 [cs]. (Visited on 14/10/2025).

[37] Thomas Schlegl et al. 'Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery'. In: *Information Processing in Medical Imaging*. Ed. by Marc Niethammer et al. Cham: Springer International Publishing, 2017, pp. 146–157. ISBN: 978-3-319-59050-9. DOI: 10.1007/978-3-319-59050-9_12.

[38] Dong Gong et al. 'Memorizing Normality to Detect Anomaly: Memory-Augmented Deep Autoencoder for Unsupervised Anomaly Detection'. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, Korea (South): IEEE, Oct. 2019, pp. 1705–1714. ISBN: 978-1-7281-4803-8. DOI: 10.1109/ICCV.2019.00179. (Visited on 14/10/2025).

[39] Thomas Schlegl et al. 'F-AnoGAN: Fast Unsupervised Anomaly Detection with Generative Adversarial Networks'. In: *Medical Image Analysis* 54 (May 2019), pp. 30–44. ISSN: 13618415. DOI: 10.1016/j.media.2019.01.010. (Visited on 14/10/2025).

[40] Thomas Defard et al. *PaDiM: A Patch Distribution Modeling Framework for Anomaly Detection and Localization*. Nov. 2020. DOI: 10.48550/arXiv.2011.08785. arXiv: 2011.08785 [cs]. (Visited on 14/10/2025).

[41] Hanqiu Deng and Xingyu Li. 'Anomaly Detection via Reverse Distillation from One-Class Embedding'. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. New Orleans, LA, USA: IEEE, June 2022, pp. 9727–9736. ISBN: 978-1-6654-6946-3. DOI: 10.1109/CVPR52688.2022.00951. (Visited on 14/10/2025).

[42] Zhihao Gu et al. 'Remembering Normality: Memory-guided Knowledge Distillation for Unsupervised Anomaly Detection'. In: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. Paris, France: IEEE, Oct. 2023, pp. 16355–16363. ISBN: 979-8-3503-0718-4. DOI: 10.1109/ICCV51070.2023.01503. (Visited on 14/10/2025).

[43] Durk P Kingma and Prafulla Dhariwal. 'Glow: Generative Flow with Invertible 1x1 Convolutions'. In: *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc., 2018. (Visited on 28/10/2025).

[44] Eric Nalisnick et al. *Do Deep Generative Models Know What They Don't Know?* Feb. 2019. DOI: 10.48550/arXiv.1810.09136. arXiv: 1810.09136 [stat]. (Visited on 14/10/2025).

[45] Jianmei Zhong and Yanzhi Song. 'UniFlow: Unified Normalizing Flow for Unsupervised Multi-Class Anomaly Detection'. In: *Information* 15.12 (Dec. 2024), p. 791. ISSN: 2078-2489. DOI: 10.3390/info15120791. (Visited on 14/10/2025).

[46] Jiawei Yu et al. *FastFlow: Unsupervised Anomaly Detection and Localization via 2D Normalizing Flows*. Nov. 2021. DOI: 10.48550/arXiv.2111.07677. arXiv: 2111.07677 [cs]. (Visited on 14/10/2025).

[47] Arian Mousakhan et al. 'Anomaly Detection with Conditioned Denoising Diffusion Models'. In: vol. 15297. 2025, pp. 181–195. DOI: 10.1007/978-3-031-85181-0_12. arXiv: 2305.15956 [cs]. (Visited on 14/10/2025).

[48] Matic Fučka et al. *TransFusion – A Transparency-Based Diffusion Model for Anomaly Detection*. July 2024. DOI: 10.48550/arXiv.2311.09999. arXiv: 2311.09999 [cs]. (Visited on 14/10/2025).

[49] Hang Yao et al. *GLAD: Towards Better Reconstruction with Global and Local Adaptive Diffusion Models for Unsupervised Anomaly Detection*. Sept. 2024. DOI: 10.48550/arXiv.2406.07487. arXiv: 2406.07487 [cs]. (Visited on 14/10/2025).

[50] Haoyang He et al. 'A Diffusion-Based Framework for Multi-Class Anomaly Detection'. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 38.8 (Mar. 2024), pp. 8472–8480. ISSN: 2374-3468, 2159-5399. DOI: 10.1609/aaai.v38i8.28690. (Visited on 14/10/2025).

[51] Yunkang Cao et al. 'Segment Any Anomaly without Training via Hybrid Prompt Regularization'. In: *IEEE Transactions on Cybernetics* 55.4 (Apr. 2025), pp. 1917–1929. ISSN: 2168-2267, 2168-2275. DOI: 10.1109/TCYB.2025.3536165. arXiv: 2305.10724 [cs]. (Visited on 20/10/2025).

[52] Wenxin Ma et al. 'AA-CLIP: Enhancing Zero-Shot Anomaly Detection via Anomaly-Aware CLIP'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2025, pp. 4744–4754. (Visited on 28/10/2025).

[53] Xiaofan Li et al. 'PromptAD: Learning Prompts with Only Normal Samples for Few-Shot Anomaly Detection'. In: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, June 2024, pp. 16848–16858. ISBN: 979-8-3503-5300-6. DOI: 10.1109/CVPR52733.2024.01594. (Visited on 20/10/2025).

[54] Yiyue Li et al. 'One-to-Normal: Anomaly Personalization for Few-shot Anomaly Detection'. In: *Advances in Neural Information Processing Systems* 37 (Dec. 2024), pp. 78371–78393. (Visited on 28/10/2025).

[55] Youcai Zhang et al. 'Recognize Anything: A Strong Image Tagging Model'. In: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Seattle, WA, USA: IEEE, June 2024, pp. 1724–1732. ISBN: 979-8-3503-6547-4. DOI: 10.

1109 / CVPRW63382 . 2024 . 00179. (Visited on 22/10/2025).

[56] Tianhe Ren et al. *Grounded SAM: Assembling Open-World Models for Diverse Visual Tasks*. Jan. 2024. DOI: 10 . 48550 / arXiv . 2401 . 14159. arXiv: 2401 . 14159 [cs]. (Visited on 22/10/2025).

[57] Chun-Liang Li et al. 'CutPaste: Self-Supervised Learning for Anomaly Detection and Localization'. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Nashville, TN, USA: IEEE, June 2021, pp. 9659–9669. ISBN: 978-1-6654-4509-2. DOI: 10. 1109 / CVPR46437 . 2021 . 00954. (Visited on 23/10/2025).

[58] Patrick Pérez et al. 'Poisson Image Editing'. In: *ACM Transactions on Graphics* 22.3 (July 2003), pp. 313–318. ISSN: 0730-0301, 1557-7368. DOI: 10 . 1145 / 882262 . 882269. (Visited on 23/10/2025).

[59] Olusola O. Abayomi-Alli et al. 'FruitQ: A New Dataset of Multiple Fruit Images for Freshness Evaluation'. In: *Multimedia Tools and Applications* 83.4 (Jan. 2024), pp. 11433–11460. ISSN: 1573-7721. DOI: 10.1007/s11042-023-16058-6. (Visited on 24/10/2025).

[60] J A Hanley and B J McNeil. 'The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve.' In: *Radiology* 143.1 (Apr. 1982), pp. 29–36. ISSN: 0033-8419. DOI: 10.1148/radiology.143.1.7063747. (Visited on 27/10/2025).

Part III

Closure

Conclusion

This thesis investigated reference-based anomaly detection for automated quality inspection of cluttered, multi-instance e-grocery stock containers, with the objective of detecting visible wrong products, product damage, packaging damage, and leakage from single overhead RGB images while limiting false positives caused by benign clutter and remaining feasible under warehouse deployment constraints. This chapter answers the three research questions formulated in Part I.

Research Question 1 *How can a compact and reliable SKU-specific reference set be automatically constructed for reference-based anomaly detection across a large and changing e-grocery product assortment?*

A compact and reliable SKU-specific reference set can be automatically constructed through a two-stage approach: automated VLM-based filtering removes candidate images that do not represent clean normal appearances, after which diversity-aware sampling selects a representative subset from the clean reference pool. Clean references are necessary because contaminated reference images can suppress genuine anomaly evidence during nearest-neighbour matching. Among the evaluated strategies, k-means clustering with boundary top-up yields the most favourable precision–recall operating point after false-positive mitigation, by combining coverage of dominant normal appearances with structured support for less frequent but valid product views. Reference construction therefore remains a maintenance task: packaging changes, promotional markings, and rare product orientations may require reference-pool updates or resampling over time.

Research Question 2 *How can false positives caused by product boundaries, packaging clutter, loose debris, and container artefacts be reduced while preserving sensitivity to genuine product and packaging anomalies?*

False positives can be reduced by treating raw anomaly responses as candidate evidence rather than final decisions. In cluttered stock containers, high patch-level scores may correspond to benign product boundaries, stickers, container wear, loose debris, or transport material. Edge downweighting and debris filtering address these nuisance sources by downweighting boundary-driven responses and validating high-scoring local crops before image-level decision-making. Debris filtering accounts for the largest absolute false-positive reduction and edge downweighting for the largest single F1-score improvement. The full pipeline reduces false positives from 1,414 to 201 and increases precision from 0.511 to 0.834. This precision gain comes at a recall cost, with recall decreasing from 0.893 to 0.610 (F1-score: 0.705), because the same mitigation stages can also weaken genuine anomaly evidence, particularly for leakage, open packaging, and fresh-produce defects.

Research Question 3 *How can the complete inspection pipeline remain computationally feasible under warehouse latency constraints while maintaining detection reliability?*

Computational feasibility is achieved by bounding the reference memory bank through sampling, applying debris filtering conditionally only for images that exceed the anomaly threshold, and allowing anomaly detection and product boundary extraction to run in parallel. Component selection further supports this trade-off: a compact DINOv2 backbone, YOLOv8s, MobileSAM, and ResNet-18 are selected to balance detection reliability with runtime feasibility. Product boundary extraction for edge downweighting constitutes the dominant latency contribution. The full pipeline achieves a 99th-percentile end-to-end latency of 1,403 ms, well within the operational constraint of approximately 10 s per image.

Concluding Remarks The main research question asked how a reference-based anomaly detection pipeline can be designed to detect visible anomalies in cluttered, multi-instance e-grocery stock containers while limiting false positives and remaining scalable under warehouse deployment constraints. The results indicate that this requires a modular pipeline rather than a raw anomaly detector alone: clean reference sets provide the basis for normality modelling, patch-level nearest-neighbour matching provides local anomaly evidence, and targeted false-positive mitigation prevents benign clutter from dominating the final decision. Robust inspection in this setting therefore depends not only on detecting deviations from normality, but also on deciding which deviations are operationally relevant.

A Per-SKU Dataset Composition and Results

This appendix reports the SKU-level dataset composition and performance results underlying the aggregate evaluation in Chapter 5. Table A.1 summarises the evaluation data for each of the 20 SKU classes, including reference-pool size, container diversity, evaluation-set size, and issue type. Table A.2 reports the corresponding per-SKU performance of the core detector and the full pipeline at the fixed operating thresholds defined in Section 5.1. These results are included to make the effect of SKU-specific variation, issue type, and reference availability auditable.

Table A.1: Per-SKU dataset composition across all 20 SKU classes.

SKU	Article	Issue family	Specific issue	Zone	$\lvert\mathcal{P}_s\rvert$	Pool cont.	Ref. cont.	Eval. normal	Eval. anomalous
10762165	Egg salad	Wrong-SKU	Wrong SKU	Chilled	889	37	17	150	57
10992708	Cheese sauce mix	Wrong-SKU	Wrong SKU	Ambient	1602	16	15	149	87
12140625	Chocolate muesli	Wrong-SKU	Wrong size SKU	Ambient	454	31	23	150	100
12155519	Mini potatoes†	Wrong-SKU	Wrong SKU	Chilled	562	50	21	150	100
10762910	Arla Skyr	Leakage	Yogurt and transparent leakage	Chilled	491	84	19	150	100
11299669	Semi-skimmed milk	Leakage	Milk and transparent leakage	Chilled	493	158	25	150	100
11300010	Burrata	Leakage	Transparent leakage	Chilled	676	75	17	150	100
11738660	Sour cream	Leakage	Transparent leakage; packaging puncture	Chilled	1573	63	20	150	100
12077488	Greek yogurt	Leakage	Yogurt and transparent leakage	Chilled	597	130	25	150	100
12345380	Lactose-free yogurt	Leakage	Yogurt and transparent leakage	Chilled	383	45	14	149	101
10362508	Table salt	Open packaging	Salt in container	Ambient	230	27	14	150	100
10724353	Currant buns	Open packaging	Loose bun in container	Ambient	1493	116	25	150	45
11625504	Basmati rice	Open packaging	Rice in container	Ambient	1453	132	24	150	29
11819907	Breadcrumbs	Open packaging	Breadcrumbs in container	Ambient	550	32	19	150	100
12309264	Pomegranate seeds	Open packaging	Open packagingp	Chilled	1037	80	18	150	48
90006096	Coloured carrots	Open packaging	Open packaging	Chilled	589	83	25	150	94
11406433	Avocados (4-pack)	Multipack	Out of multipack	Chilled	1010	49	23	150	100
11801970	Cola Zero (6-pack)	Multipack	Open multipack	Ambient	399	180	26	148	95
90006033	Courgette	Fresh produce	Broken product	Ambient	1031	50	20	150	68
90006068	Red pepper	Fresh produce	Spoilage; wet container	Ambient	674	47	20	150	33

Pool and ref. containers denote the numbers of distinct physical containers in $\mathcal{P}_s$ and $\mathcal{R}_s$, respectively ($\lvert\mathcal{R}_s\rvert = 30$ images per SKU for all classes). Evaluation totals include all 20 SKUs: 2,996 normal and 1,657 anomalous images. †Exceptional failure-case SKU: the anomaly comprises an almost visually identical product variant with unchanged packaging. This SKU is included in the table totals but excluded from aggregate results marked with $*$ in the main text (see Section 5.2).

Table A.2: Per-SKU performance across all 20 SKU classes. The core detector is summarised by F1-score, while the full pipeline is reported using false positives, false negatives, precision, recall, and F1-score. The fixed operating thresholds are defined in Section 5.1.

			Core det.	Full pipeline				
SKU	**Article**	**Issue family**	**F1**	**FP**	**FN**	**Prec.**	**Rec.**	**F1**
10762165	Egg salad	Wrong-SKU	0.720	4	20	0.902	0.649	0.755
10992708	Cheese sauce mix	Wrong-SKU	0.849	15	6	0.844	0.931	0.885
12140625	Chocolate muesli	Wrong-SKU	0.712	4	1	0.961	0.990	0.975
12155519	Mini potatoes†	Wrong-SKU	0.139	1	99	0.500	0.010	0.020
10762910	Arla Skyr	Leakage	0.604	30	20	0.727	0.800	0.762
11299669	Semi-skimmed milk	Leakage	0.601	15	29	0.826	0.710	0.763
11300010	Burrata	Leakage	0.618	3	67	0.917	0.330	0.485
11738660	Sour cream	Leakage	0.645	15	60	0.727	0.400	0.516
12077488	Greek yogurt	Leakage	0.683	10	29	0.877	0.710	0.785
12345380	Lactose-free yogurt	Leakage	0.614	16	38	0.798	0.624	0.700
10362508	Table salt	Open packaging	0.641	54	16	0.609	0.840	0.706
10724353	Currant buns	Open packaging	0.756	0	16	1.000	0.644	0.784
11625504	Basmati rice	Open packaging	0.495	0	12	1.000	0.586	0.739
11819907	Breadcrumbs	Open packaging	0.678	2	38	0.969	0.620	0.756
12309264	Pomegranate seeds	Open packaging	0.781	0	20	1.000	0.583	0.737
90006096	Coloured carrots	Open packaging	0.651	19	44	0.725	0.532	0.614
11406433	Avocados (4-pack)	Multipack	0.733	2	51	0.961	0.490	0.649
11801970	Cola Zero (6-pack)	Multipack	0.775	11	7	0.889	0.926	0.907
90006033	Courgette	Fresh produce	0.511	0	50	1.000	0.265	0.419
90006068	Red pepper	Fresh produce	0.635	0	23	1.000	0.303	0.465
Overall			**0.650**	**201**	**646**	**0.834**	**0.610**	**0.705**

The overall row includes all 20 SKUs. Aggregate results excluding the exceptional failure-case SKU are reported with $*$ in the main text (see Section 5.2).

B Inference Configuration

This appendix is included to support reproducibility by recording the model variants and inference parameters used to produce the evaluation results reported in Chapter 5. Input resolutions, hardware configuration, and the general software stack are described in Section 5.1. Training settings for the two custom-trained components (YOLO object detector and ResNet-18 debris classifier) are provided in Appendix C.

Table B.1: Model variants used at inference across all pipeline stages.

Component	**Model**	**Training status**
Feature extraction	DINOv2 ViT-S/14 [6]	Frozen
Object detection	YOLOv8s [25]	Custom trained
Instance segmentation	MobileSAM [26]	Frozen
Debris classification	ResNet-18 [27]	Custom trained
Variant verification	EasyOCR [24]	Frozen

Table B.2: Background masking parameters by container type.

Parameter	Chilled	Ambient	Crate
Trapezoidal ROI			
Top lateral margin (each side)	15.5%	15.5%	19.0%
Bottom lateral margin (each side)	6.9%	7.8%	13.8%
Top ROI height fraction	4.5%	4.5%	13.4%
Bottom ROI height fraction	4.0%	4.0%	5.5%
CLAHE enhancement			
Clip limit	2.0	4.0	4.0
Tile grid size	10×10	12×12	12×12
Canny edge detection			
Gaussian blur kernel		5×5	
Adaptive sigma		0.50	
Probabilistic Hough transform			
Accumulator threshold		50	
Horizontal min. line length		W / 8	
Horizontal max. line gap		20 px	
Vertical min. line length		H / 6	
Vertical max. line gap		12 px	
Vertical line angle tolerance		$90^\circ \pm 4^\circ$	
Horizontal line angle tolerance		$< 2^\circ$	

ROI denotes the region of interest used for container-rail line detection. Percentages are defined relative to the pre-cropped image dimensions. W and H denote the width and height of the pre-cropped image, respectively.

Table B.3: AnomalyDINO inference parameters.

Parameter	Value
Backbone	`dinov2_vits14`
Input resolution (smaller edge)	448 px
Reference set size $\lvert\mathcal{R}_s\rvert$	30 images per SKU
k-nearest neighbours	1
Feature normalisation	L2-normalised
Preprocessing mode	`agnostic`
Random seed	0

The `agnostic` preprocessing mode applies the Hough-based background masking detailed in Table B.2.

Table B.4: Inference parameters for object detection, instance segmentation, and edge downweighting.

Parameter	Value
Object detection (YOLOv8s)	
Detection confidence threshold	0.30
Input image size	640×640 px
Instance segmentation (MobileSAM)	
Prompt mode	Bounding box
Internal long-side resize	1024 px
Minimum mask area	400 px^2
Minimum mask compactness	0.01
Maximum mask area fraction	0.95
Deduplication IoU threshold	0.98
Mask score threshold	-0.30
Connected-component splitting	Enabled (min. 200 px^2)
Edge downweighting	
Outward edge radius	12 px
Outward suppression weight	0.40
Inward edge radius	12 px
Inward suppression weight	0.40

The product boundary extraction step runs in parallel with AnomalyDINO. Edge downweighting is applied after both branches complete. Edge radii and mask-area thresholds are defined in the pre-cropped image space and projected to the anomaly-map patch grid by area-averaged downsampling.

Table B.5: Inference parameters for candidate crop extraction and debris filtering.

Parameter	Value
Candidate crop extraction	
Image-level anomaly threshold	0.230
Patch-level anomaly threshold	0.245
Top-K crops per image	7
Minimum crop area (patches)	2
Crop padding ratio	0.00
Crop mask dilation iterations	1
Wall filter	
Patch margins (top / bottom / back / crate)	2 / 3 / 5 / 2
Overlap threshold	0.50
Reflection filter (HSV colour space)	
Minimum brightness value (V)	170
Maximum saturation (S)	60
Bright-pixel area threshold	0.20
Debris filter (ResNet-18)	
Classification threshold	0.15
Input crop size	224×224 px

The wall and reflection filters are applied during crop extraction and suppress nuisance crops before debris classification. Top-K caps the number of candidate crop regions forwarded to the debris filter per image, with regions selected in descending patch-anomaly score order.

Table B.6: Inference parameters for variant OCR verification using EasyOCR.

Parameter	Value
Rotation angles tested	0°, 90°, 180°, 270°
Target image resize (long side)	600 px
Minimum non-reference token length	5 characters
Non-reference token confidence threshold	0.70
Fuzzy edit distance (Levenshtein)	2
Maximum reference images used	30

Non-reference tokens denote OCR-detected text elements in the query image that do not match any token in the SKU-specific reference vocabulary within the specified edit distance.

C Auxiliary Model Training Configurations

This appendix is included to support reproducibility by documenting the training configuration for the two custom-trained auxiliary components of the pipeline. Both models are initialised from pretrained weights and fine-tuned on domain-specific data. The corresponding inference-time operating parameters are reported in Appendix B.

Table C.1: YOLOv8s object detector training configuration.

Parameter	Value
Dataset	
Training images	400
Validation images	113
Test images	54
Classes	1 (item)
Annotation type	Bounding boxes
Architecture and initialisation	
Architecture	YOLOv8s
Pretrained weights	COCO
Optimisation	
Optimiser	SGD (`optimizer=auto`)
Initial LR (lr_0)	0.01
Final LR factor (lrf)	0.01
Momentum	0.937
Weight decay	5×10^{-4}
Warmup epochs	3
Batch size	32
Max epochs	150
Early stopping	Patience 300 (disabled)
AMP	Enabled
Seed	0
Data augmentation	
Augmentation policy	YOLOv8 defaults (unmodified)
Image size	640×640 px

YOLOv8 training augmentation settings were left at the Ultralytics defaults. Trained using Ultralytics `8.4.21`.

Table C.2: ResNet-18 debris classifier training configuration.

Parameter	Value
Dataset	
Training set	7,041 crops (5,264 debris; 1,777 not-debris)
Validation set	1,760 crops (1,316 debris; 444 not-debris)
Split strategy	Stratified 80/20
Architecture and initialisation	
Architecture	ResNet-18
Pretrained weights	ImageNet
Input crop size	224×224 px
Output classes	2 (debris, not-debris)
Optimisation	
Optimiser	AdamW
Learning rate	1×10^{-4}
Weight decay	1×10^{-4}
Batch size	64
Max epochs	30
LR scheduler	ReduceLROnPlateau (factor 0.5, patience 2)
Early stopping	Patience 15, min. ΔF1 = 0.002
Model selection metric	Validation F1-score
Loss function	Weighted cross-entropy
Seed	202
Data augmentation	None

D VLM Reference Filtering Configuration

This appendix documents the vision-language model configuration used for automated image-level filtering during reference pool construction, as described in Subsection 4.2.1. No model fine-tuning is performed. Candidate reference images are classified through zero-shot prompting and structured output parsing. The tables report the model routing, input structure, arbitration logic, and SKU metadata used for prompt construction. A condensed version of the leakage-specific prompt is included as a representative example to illustrate the prompt structure and decision criteria. The standard and multipack prompts follow the same structured-output format but use different SKU metadata injections and task-specific rules.

Table D.1: VLM filtering configuration for reference pool construction.

Parameter	Value
Task routing	
Standard and multipack inspection	GPT-4.1
Leakage inspection	GPT-5.1
Inference settings	
Temperature	0.0
Max. concurrent API requests	15
Image encoding	PNG (lossless)
Image inputs	
Standard and multipack task	Template SKU image + container image
Leakage task	Container image + top and bottom floor-tile crops
Arbitration	
Execution	Parallel when multiple task branches apply
Decision rule	ANOMALOUS if any applicable task returns ANOMALOUS

GPT-5.1 was used for leakage inspection to apply a stricter leakage-specific prompt to subtle liquid-residue cases. The standard prompt is dynamically adapted with SKU-specific product metadata (name, brand, packaging type, content). The multipack prompt follows the same structure with additional bundle-integrity rules.

Table D.2: SKU metadata fields used for VLM filtering during reference pool construction.

Field	Role in prompt
	Direct prompt injections
Product name	Expected SKU name, provided as textual context
Content summary	Expected product contents and weight/volume
Packaging type	Expected packaging; activates packaging rules
	Conditional rule activators
Brand name	Selects branded or generic brand-matching rule
Category	Activates category rules (fruit & veg, deli tray exception)
Container identifier	Detects crate vs. tote; activates crate moisture exception
Leakage flag	Routes image to the parallel leakage inspection task
Multipack flag	Selects multipack prompt template over standard template
Multipack containment flag	Activates strict unit-containment rules

Packaging type serves two functions: it is injected as a named field and also used to activate packaging-specific rule blocks (sleeve, gable top, cup/tub, tray). Product metadata are supplied per SKU rather than hard-coded in the prompt templates.

```
# Role and Objective
You are a Forensic Surface Analyst. Detect PRODUCT LEAKAGE (liquids, smears, residues) while ignoring benign
     artefacts such as specular reflections, packaging colour casts, and dry material scuffs.

## Discrimination Guide
WET vs. REFLECTION: Normal reflections are sharp, linear, and follow tote geometry. Leakage alters surface
     texture (diffused gloss, smearing) and shows pooling, droplets, or residue boundaries.

WHITE MARK: Dry scuffs are matte and chalky (Normal). Leakage residue is glossy, creamy, translucent, or
     viscous (Anomalous).

CARDBOARD: Normal shadows are diffuse and unsaturated. Wicking is high-contrast with feathered edges,
     saturated or bleeding into fibres (Anomalous).

{cup_rule}

## Decision Logic
Classify ANOMALOUS if one or more leakage indicators are present and cannot be reasonably explained by
     benign artefacts: physical volume, pooling, droplets, smear streaks, residue boundaries, broken
     reflection patterns, or saturated cardboard wicking. A glossy patch alone requires at least two
     independent liquid-behaviour cues.
Classify NORMAL if marks are matte/chalky, consistent with specular highlights, or are diffuse shadows
     without saturation.

## Final Rules
If the appearance is fully explained by lighting, reflections, packaging colour casts, or dry scuffs,
     classify NORMAL.
{cup_final_arb}
Do not classify as ANOMALOUS based solely on irregular shape; curved plastic naturally creates irregular
     reflections.
{cup_final}

## Output Format (JSON only)
{
  "status": "NORMAL" or "ANOMALOUS",
  "reason": "..."
}
```

Listing D.1: Condensed representative leakage inspection prompt used for VLM-based reference filtering.

The listing is condensed for presentation. The operational prompt includes additional checklist steps. The placeholders `{cup_rule}`, `{cup_final_arb}`, and `{cup_final}` are filled at runtime for cup or tub products with additional anti-false-positive rules for transparent-base packaging and are empty strings for all other packaging types. Only the `status` field drives automatic filtering. The `reason` field is retained for traceability.

E Literature Review

This chapter reviews the existing literature relevant to automated quality inspection using visual anomaly detection. It analyses current approaches, datasets, and evaluation practices to form the theoretical and methodological basis for this thesis. First, it introduces the main methodological families within unsupervised anomaly detection, followed by an assessment of recent state-of-the-art methods and their suitability for tote inspection. The chapter concludes with an overview of benchmark datasets and evaluation metrics, followed by a discussion of the research gaps derived from this analysis.

E.1. Overview of Unsupervised Anomaly Detection Approaches

Automated defect detection in multi-instance tote images relies on identifying deviations from normal appearance without access to extensive labelled data. Because labelled defect samples are rarely available, most recent approaches are developed within an unsupervised anomaly detection framework, learning the characteristics of normal visual patterns and detecting deviations from them. This section provides a conceptual overview of the main modelling families underpinning current research. The following subsections outline reconstruction-based, representation- and memory-based, flow-based, diffusion-based, and vision–language-model approaches, summarising their core principles, advantages, and limitations. These foundations establish the analytical context for the subsequent discussion of state-of-the-art methods tailored to the tote inspection problem.

E.1.1. Reconstruction-Based Methods

Reconstruction-based methods for anomaly detection rely on the principle that models trained only on normal data will reconstruct those patterns accurately, while failing to reproduce unseen or abnormal inputs. These methods are typically implemented using unsupervised generative architectures such as Autoencoders (AEs), Variational Autoencoders (VAEs), and Generative Adversarial Networks (GANs) [34–37]. These models generally consist of an encoder–decoder pair, where the encoder maps the input into a compact latent representation and the decoder reconstructs it back into the input space. The network is trained to minimise the reconstruction error on normal samples without anomalies [8, 38]. At inference, the discrepancy between the input and its reconstruction is quantified, and inputs with significantly higher reconstruction error are flagged as anomalous [34, 38, 39]. The underlying assumption is that anomalies lie outside the learned representation of normal data and therefore cannot be efficiently encoded or reconstructed [35].

Reconstruction-based approaches offer several advantages for defect detection in settings where labelled data are scarce. Because these models learn normality only from non-defective samples, they can operate effectively without labelled defect data and remain agnostic to specific defect types [34, 35]. In addition, reconstruction-based frameworks produce pixel-wise difference maps that highlight regions that deviate from the learned representation of normal appearance, providing clear visual cues for localising potential defects within complex and cluttered scenes, such as multi-instance product totes [8, 38, 39].

Despite these advantages, reconstruction-based models face several limitations when applied to complex, real-world inspection scenarios. First, these methods typically require a large and diverse set of normal samples to learn a consistent representation of normal appearance. When normal data are limited or highly variable, such as the assortment of products within totes, the learned model may either underfit or over-generalise, resulting in poor defect sensitivity [36, 39]. Second, the presence of multiple, overlapping objects within a single image introduces an over-generalisation effect, the network tends to reconstruct the entire scene rather than individual items. As a result, small or localised defects within otherwise normal scenes may be overlooked, as the overall reconstruction error remains dominated by normal regions [8, 38]. Third, balancing model capacity is critical. A network with excessive representational power may reconstruct even defective regions accurately, suppressing anomaly signals, whereas an overly constrained model increases false positives. Memory-augmented designs such as MemAE address this by limiting reconstruction to prototypical normal patterns [38]. DRÆM further improves sensitivity to subtle anomalies by coupling reconstruction and discrimination, encouraging the model to amplify deviations instead of smoothing them out [8]. Finally, computational efficiency remains a concern in time-sensitive inspection pipelines. Early GAN-based detectors like AnoGAN required iterative optimisation to project test images into latent space, resulting in slow inference incompatible with real-time operation. The f-AnoGAN framework mitigates this by introducing an encoder for direct latent mapping, thereby reducing inference time while maintaining accuracy [39]. Consequently, although these methods provide pixel-level localisation, their high data requirements, tendency to over-generalise in multi-instance scenes, and relatively slow inference make them an impractical foundation for the data-scarce and real-time tote inspection problem.

E.1.2. Representation- and Memory-Based Methods

Representation- and memory-based methods identify anomalies by comparing an image's deep feature representations to the learned representation of normality, rather than reconstructing it in pixel space. These approaches operate in the latent feature space of convolutional neural networks (CNNs), typically pre-trained on large datasets such as ImageNet, which extract multi-level features that capture both local textures and global structural patterns [13, 40]. The core assumption is that features extracted from normal samples form a coherent region in this feature space, while abnormal regions produce feature vectors that differ significantly from it [13, 40]. Normality can be represented explicitly, by storing patch-level feature embeddings or modelling their statistics through probabilistic distributions [3, 40], or implicitly, through knowledge distillation between a pre-trained teacher network and a student trained only on normal data [7, 41]. During inference, features from a test image are compared to this learned representation using distance-based metrics such as k-nearest neighbour or Mahalanobis distance, or by evaluating discrepancies between teacher and student representations [13, 41]. Image regions that deviate strongly in this feature space are then flagged as anomalous, allowing precise localisation of potential defects [3, 18].

Representation- and memory-based methods offer several advantages for visual anomaly detection. They require only normal samples for training and achieve strong performance even with limited data, as shown by DN^2 and PatchCore, which operate effectively with as few as tens of normal images per class [3, 13]. This data efficiency results from their use of pre-trained feature extractors rather than end-to-end optimisation, allowing new products to be introduced without extensive retraining. Because these models rely on general-purpose visual features learned from large-scale datasets such as ImageNet, they can adapt to a wide range of product types without class-specific fine-tuning [3, 13, 40]. By analysing features at patch level, they can identify small or localised defects within complex scenes. SPADE and PatchCore combine multi-scale contextual matching to achieve

accurate localisation across diverse surface patterns [3, 18]. Once the representation of normality is established, methods such as PaDiM and PatchCore perform inference efficiently, replacing iterative reconstruction with direct feature comparison or Gaussian modelling [3, 40].

Despite these strengths, several challenges arise when applying representation- and memory-based methods to multi-instance tote images with limited normal data. These approaches assume that the stored feature distribution accurately represents normal appearance. Meaning that unseen orientations or partial occlusions of products, common in tote images, can lead to false anomaly detections, as they fall outside learned the feature distribution [13, 18]. Explicit memory-based frameworks such as DN^2, SPADE, and PatchCore also scale poorly with the number of stored features. Inference time and memory requirements increase linearly with dataset size [3, 13, 18, 40]. While PatchCore addresses this issue through coreset-based feature reduction, this comes at the cost of a small accuracy loss and requires the memory bank to be updated whenever new SKUs are introduced [3]. Finally, knowledge-distillation approaches such as RD4AD and MemKD, although more compact, can over-generalise during training, causing the model to reproduce anomalous features as normal and resulting in missed detections [41, 42]. Reliable performance in grocery tote inspection therefore depends on adequate coverage of normal variation, scalable feature memory, and robustness to orientation and clutter.

E.1.3. Flow-Based Methods

Flow-based methods for anomaly detection model the distribution of normal data using normalising flows (NFs), a class of invertible neural networks that explicitly represent probability distributions [14, 43, 44]. An NF defines a bijective mapping between the input space and a latent space, enabling the probability density of each sample to be computed exactly through the change-of-variables formula [43–45]. The model is trained exclusively on normal samples, learning transformations that map these images into a latent space following a simple, tractable distribution, typically a multivariate standard normal [14, 43]. Because the transformation is fully invertible, its likelihood can be computed directly from the latent representation, and the input features provided to the flow can be reconstructed precisely. During inference, the model estimates how likely a new image is to belong to the learned normal distribution, and samples with unusually low likelihoods are identified as anomalous [14, 44].

Flow-based methods combine several properties that make them effective for real-time visual anomaly detection. Like representation-based approaches, they require only a few normal samples and achieve both efficient inference and precise anomaly localisation by modelling the probability distribution of normal features directly [14, 46]. However, unlike memory-based techniques, their inference time remains constant regardless of training data size, making them more scalable to large and evolving product assortments [9]. A further advantage is their robustness to variations in object orientation and placement, through invertible feature mappings and data augmentation, methods such as DifferNet and UniFlow maintain stable anomaly scores under diverse viewing conditions [14, 45]. This robustness is particularly beneficial for tote inspection, where products appear in random poses and partial occlusions are common.

Despite their advantages, flow-based models face several challenges when applied to tote inspection. First, the reliability of likelihood as an anomaly indicator remains a limitation. Flow models estimate probability density directly, yet may assign higher likelihoods to out-of-distribution inputs than to true normal samples [44]. This comes from their focus on low-level statistics rather than semantic structure, which can cause defective products to be misclassified as normal when surface textures resemble other non-defective items. Feature-based approaches such as DifferNet and CFLOW-AD reduce this effect by applying flows to deep features instead of pixels [9, 14], but likelihood scores still require careful calibration. Second, modelling a single normal distribution across visually diverse products increases false positives, as normal items with distinct appearances may fall outside the learned manifold. Standard flows assume all samples belong to one distribution, limiting their ability to represent multiple product types [14]. Training a model per SKU could address this but is infeasible at scale, while recent methods such as UniFlow attempt to mitigate this through enhanced learning capacity and feature adaptation to model multi-class distributions effectively [45]. Finally, the effectiveness of flow-based models depends on sufficient normal data to capture valid variations in pose and appearance. With few or no normal tote images for many products, even legitimate orientations or packaging differences may be flagged as anomalies [14, 45]. Therefore, while their constant inference time and inherent pose robustness align well with the tote inspection requirements, their challenges with likelihood calibration and the need for sufficient normal data to model diverse poses and SKUs limit their practical suitability. Overall, the constant inference time and pose robustness achieved through data augmentation align well with tote inspection requirements, but challenges in likelihood calibration and the use of a single normal distribution across visually diverse products currently limit their practical applicability.

E.1.4. Diffusion-Based Methods

Diffusion-based methods for visual anomaly detection learn the distribution of normal images through a gradual noising–denoising process that reconstructs anomaly-free counterparts of defective inputs. These approaches are built upon Denoising Diffusion Probabilistic Models (DDPMs) and related latent or conditional formulations. They learn to corrupt normal samples by incrementally adding Gaussian noise and then reverse this process by progressively denoising them back to the original distribution [10, 15]. Trained exclusively on normal data, the model learns to represent normal appearance and applies this knowledge during inference to restore potentially defective images to their normal form. It does so by removing structures that fall outside the learned distribution. As a result, anomalous regions are reconstructed as normal while defect-free areas are preserved. Anomalies are then detected and localised by comparing the reconstructed image with the input through pixel-wise or feature-wise differences [47–49].

Diffusion-based methods have several properties that make them effective for unsupervised defect detection in complex, cluttered tote environments. Through iterative denoising, they model image structure across multiple spatial scales, allowing reliable detection of both fine surface defects and large structural deviations within a single framework [10, 15, 49]. This robustness to anomaly scale is particularly useful for tote inspection, where defects may range from small scratches on individual items to missing or deformed products among overlapping objects [48, 49]. Diffusion frameworks also scale well to multi-class inspection tasks, as models such as DiAD and DDAD learn shared semantic representations that improve generalisation across visually diverse products [47, 50]. Although some adaptation to new product types remains necessary, this shared modelling of normal appearance across categories supports more efficient deployment in large and evolving product assortments typical of tote inspection.

Despite their high reconstruction quality, diffusion-based models face several limitations when applied to real-time tote inspection. First, normal tote images per product are scarce, which poses a major limitation for diffusion-based models that require large sets of normal samples for stable learning [47, 49, 50]. Maintaining semantic consistency in cluttered scenes also remains difficult. Diffusion models may unintentionally alter the appearance or orientation of normal products during reconstruction, leading to false anomaly localisation in multi-instance totes [15, 50]. Conditioning strategies such as those used in DDAD and DiAD help mitigate this issue but also increase model complexity [47, 50]. Second, diffusion models are computationally demanding, as generating anomaly-free reconstructions requires many iterative denoising steps, making inference slower than flow- or representation-based approaches [10, 15, 49]. Methods such as AnoDDPM, DiffAD, and DDAD-S reduce runtime through partial diffusion, latent-space processing, or compressed model designs, but these improvements often decrease reconstruction quality for large or complex defects [10, 15, 47]. Third, diffusion models can still experience over-reconstruction, where small or subtle defects are reproduced too precisely and therefore missed during detection [15, 49]. Recent methods such as GLAD and TransFusion mitigate this through anomaly-oriented training and guided denoising, but ensuring that subtle defects are diffused correctly remains a key challenge for tote inspection [48, 49]. Finally, modern diffusion frameworks such as DiAD and TransFusion employ multi-stage architectures with several subnetworks and loss functions, increasing integration complexity and reducing computational efficiency [48, 50]. Ultimately, diffusion-based models achieve high reconstruction fidelity and robust multi-scale, multi-class detection, but their strong data dependence and high computational latency from iterative denoising make them unsuitable for real-time tote inspection under data-scarce conditions.

E.1.5. Vision-Language-Model Methods

Vision-Language-Model (VLM) methods for anomaly detection build on large-scale pre-trained models capable of recognising visual concepts without task-specific training [12]. Models such as CLIP establish a shared embedding space between images and text, allowing semantic comparison between visual content and descriptive prompts [12, 16]. By relating image features to textual descriptions of normal and defective states, these methods can perform zero-shot or few-shot anomaly detection using natural language prompts instead of large labelled datasets [11, 16, 51]. During inference, the similarity between visual embeddings and text-based representations of “normal” and “anomalous” concepts determines the anomaly score [11, 16, 52]. This capability allows the detection of new or unseen defects through the general visual understanding of the model, reducing the need for product-specific retraining or defect-specific model design [12, 16].

VLM methods offer several advantages that make them particularly suitable for data-limited and class-agnostic inspection scenarios. Pre-trained on large-scale image–text data, models such as CLIP provide broad visual and semantic knowledge that enables anomaly detection without task-specific training [11, 12, 16]. This capability allows for zero- or few-shot operation using only normal images or descriptive prompts, addressing the scarcity of labelled normal and defective data in tote inspection [11, 16, 51, 53]. WinCLIP demonstrates that a pre-trained CLIP can classify and localise industrial defects without fine-tuning, relying only on prompts describing “normal” and “anomalous” states [11]. Similarly, One-to-Normal uses few-shot normal images to personalise anomaly detection for unseen products, showing strong adaptability to new SKUs [54]. Such generalisation across unseen classes provides a clear advantage over reconstruction- and flow-based methods, which typically require product-specific training to model normality [11, 12, 51]. Another strength is their semantic flexibility. By linking visual and textual concepts in a shared embedding space, VLMs interpret prompts that describe general or specific defect conditions, providing contextual guidance unavailable to purely visual models [12, 16, 53]. This language interface enables prompt-based adaptation, allowing cues such as “intact surface” or “damaged component” to focus detection on relevant regions while ignoring minor variations [11, 51, 52].

Despite their strong generalisation ability, Vision-Language-Model methods face several limitations when applied to tote inspection. First, pre-trained models such as CLIP are not optimised for fine-grained defect detection, as they focus on global object semantics rather than subtle local irregularities [12, 16, 52]. This can cause small scratches or deformations to be overlooked, especially in cluttered scenes with many overlapping items like in multi-instance tote images [11, 16, 52]. Recent methods such as AnomalyCLIP address this limitation by learning object-agnostic prompts that redirect attention from product identity to visually abnormal regions, improving sensitivity to local defects while maintaining class-agnostic generalisation [11, 16, 52]. Second, accurate localisation remains difficult. Since VLMs are trained for image-level recognition, they focus on global embeddings and output a single similarity score rather than pixel-wise anomaly maps, making it difficult to localise defective regions [11, 16]. Methods such as WinCLIP and SAA+ introduce local feature extraction or segmentation modules, using windowed patch matching or SAM integration, but these additions increase computational complexity [11, 16, 51]. Third, detection reliability is highly sensitive to prompt design. Naive or overly generic text descriptions often lead to false detections or inconsistent results [51]. In addition, the need for product-specific or manually engineered prompts limits scalability [16, 53]. Adaptive prompt-learning methods such as PromptAD and AnomalyCLIP mitigate this dependence by learning contextual or object-agnostic prompts, but effective prompting remains a key challenge in diverse tote environments [16, 51, 53]. Finally, the combined use of large transformer backbones, multi-prompt inference, and patch-based processing increases computational demands and hinders real-time performance [11, 12, 16]. Even optimised frameworks such as WinCLIP and One-to-Normal remain slower than lightweight approaches like flow-based models, with measurable inference latency reported for both methods [11, 54].Their zero-shot generalisation and adaptability make VLMs highly valuable for large and frequently changing product assortments, but their reliance on prompt engineering, limited fine-grained localisation, and high computational demands currently constrain their use in real-time tote inspection.

E.1.6. Comparative Analysis and Relevance to Tote Inspection

When data are scarce and defects are unseen across a large SKU set, representation- and memory-based methods and flow-based detectors are the strongest starting points. Both learn from few normal images and remain class agnostic by operating in feature space. Representation methods leverage pre-trained backbones and offer precise patch-level localisation, although feature banks must be controlled to limit memory growth and drift. Flow models are compact and training-light, yet likelihood calibration and the representation of multi-modal normality remain open issues. Vision–language-model methods aid zero- or few-shot onboarding across many SKUs, but they are prompt sensitive and weaker at fine-grained localisation.

In multi-instance totes with clutter, occlusion and random pose, representation-based methods and flows provide effective patch-level localisation while being efficient with compact designs. Reconstruction and diffusion approaches can produce detailed maps,

but they typically need larger normal datasets and are slower in practice, which limits deployment. VLM pipelines often require additional modules for localisation that increase runtime and still fall short on precision. Overall, compact representation- and memory-based models, optionally paired with flow scoring on deep features, offer the best balance across the four core challenges, making them the primary foundation for the methods developed in this thesis. VLM cues can complement these by improving class-agnostic onboarding and attention. The next section on state-of-the-art methods translates these conclusions into concrete architectures and hybrid designs for tote inspection.

E.2. State-of-the-Art Methods

Recent state-of-the-art (SOTA) approaches in anomaly detection increasingly adopt training-free and few-shot frameworks that tackle data scarcity without requiring defect annotations or SKU-specific retraining. These methods combine powerful pre-trained visual and vision–language models with efficient patch-level or multimodal reasoning to enhance generalisation and localisation across diverse product types. This section reviews three representative approaches, AnomalyDINO [4], UniVAD [19] and AnomalyGPT [5], which respectively extend representation-, hybrid-, and vision–language-based paradigms to enable scalable and class-agnostic tote inspection under realistic industrial conditions. These methods serve as primary experimental baselines and conceptual references for the framework developed in this thesis.

E.2.1. AnomalyDINO

AnomalyDINO [4] introduces a training-free, patch-based framework for one- and few-shot industrial anomaly detection. It builds upon the non-parametric memory logic of PatchCore [3] and deep kNN methods like DN^2 [13], but replaces earlier self-supervised backbones with DINOv2 [6] to leverage stronger, general-purpose visual features.

The overall process is illustrated in Figure E.1. From one or a few clean reference samples, the model extracts patch embeddings $f(x) = (p_1, \ldots, p_n)$ that together form a nominal memory $\mathcal{M}$. During inference, a test image x_{test} is encoded in the same way and each test patch is compared to its nearest neighbour in $\mathcal{M}$ using the distance function $d_{\text{NN}}(p; \mathcal{M})$. This *NN-matching* step yields a set of patch distances, which are upsampled using bilinear interpolation and Gaussian smoothing to the original image resolution to produce a pixel-level anomaly map. The highest patch distances are then aggregated into a single image-level anomaly score $s(x_{\text{test}})$ using a statistic q, defined as the mean of the top 1% most anomalous patches. To improve robustness in few-shot settings, AnomalyDINO includes an optional foreground masking step that estimates which regions belong to the object using principal component analysis (PCA) on DINOv2 features, thereby reducing the influence of background clutter. Reference images can also be rotated to add normal orientation variants to the memory. Together, these components allow AnomalyDINO to achieve strong one- and few-shot results on MVTec-AD [1] and VisA [2] without any fine-tuning or class supervision.

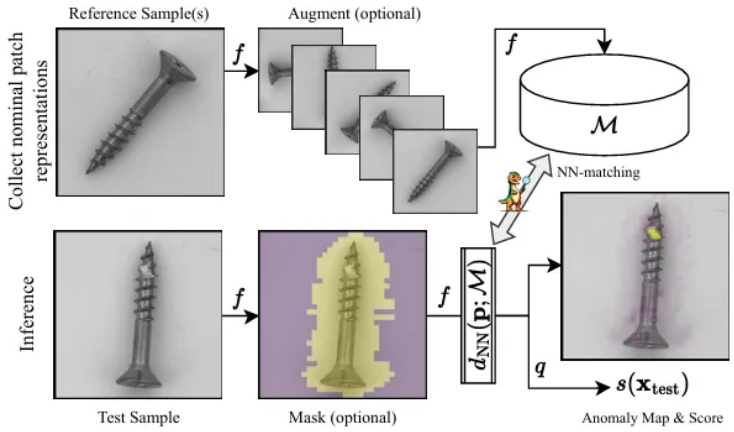


Figure E.1: AnomalyDINO architecture adapted from [4].

When applied to the tote-based inspection problem, AnomalyDINO offers a promising yet challenging foundation. Its use of DINOv2 features enables good generalisation from very few normal examples, which partially mitigates data scarcity and supports class-agnostic detection across many SKUs. However, the method's non-parametric memory $\mathcal{M}$ requires a comprehensive representation of normal appearance. In the tote setting, products can appear in a wide range of orientations, scales, and occlusions, making it difficult to capture all variations with only one or a few reference images. As a result, unseen normal poses or partial views may be incorrectly scored as anomalies. Although the optional zero-shot masking step helps to suppress background regions, accurate masking remains difficult in multi-instance scenes where products overlap or share visual similarity with the tote or packaging materials. In such cases, the mask may incorrectly exclude valid object regions or include irrelevant areas such as cardboard. Despite these challenges, AnomalyDINO remains computationally efficient: a single DINOv2 ViT-S forward pass and matrix-based nearest-neighbour matching can operate within real-time constraints on a GPU. Overall, its success for tote-level inspection depends on achieving representative normal coverage in the memory and reliable masking under complex, cluttered conditions. These challenges may be mitigated through tailored data augmentations or adaptive masking strategies.

E.2.2. UniVAD

UniVAD [19] is a unified framework for few-shot visual anomaly detection that generalises across industrial, logical, and medical domains. It eliminates the need for category-specific training by leveraging pre-trained vision and vision–language foundation

models. During inference, only a small set of normal reference images is required. Its novelty lies in combining multi-level feature reasoning, from local textures to global component structure, within a single architecture. This unified design enables anomaly detection without retraining and enhances generalisability and scalability to previously unseen SKUs and domains.

UniVAD's inference pipeline (Figure E.2) consists of three main modules: Contextual Component Clustering (C3), Component-Aware Patch Matching (CAPM), and Graph-Enhanced Component Modeling (GECM). The C3 module first performs component-level segmentation using a hybrid of the Recognize Anything Model (RAM) [55] and Grounded SAM [56] to generate coarse object masks. These masks are then refined through K-means clustering applied to features extracted by a frozen image encoder. This enables consistent segmentation of objects or components under few-shot conditions.

CAPM computes patch-level anomaly scores by comparing features from query and reference images (P_q, P_n) within each segmented component. By restricting nearest-neighbour matching to corresponding components, CAPM reduces false positives in multi-instance scenes. It also forms an image–text score by comparing patch features with text prompts that describe normal and anomalous states. In parallel, GECM extends this reasoning to the component level. It constructs a graph whose nodes represent components and applies graph attention to aggregate inter-component context. Each component then receives a component-level anomaly score based on its distance to the closest normal component embedding. This deep score is combined with geometric cues such as area, position, and colour to detect missing, added, or misplaced items. Finally, UniVAD fuses the patch-level structural map and the component-level logical map into a unified anomaly map and global score. This multi-scale reasoning enables detection of both local defects and global compositional anomalies in a single forward pass.

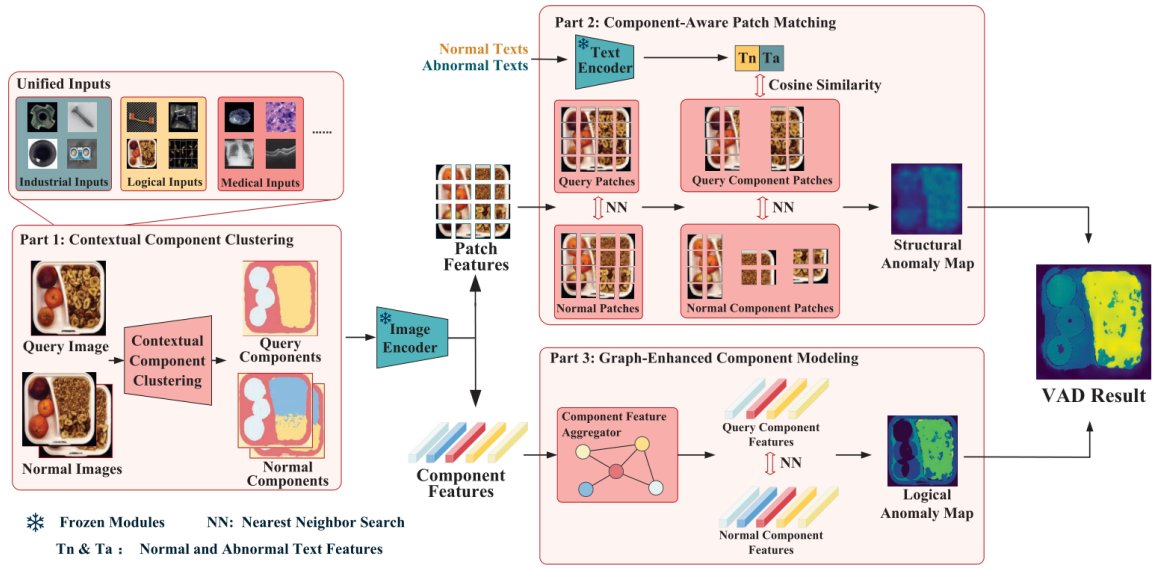


Figure E.2: Overview of the UniVAD architecture adapted from [19].

For multi-instance tote inspection, UniVAD presents several relevant properties and challenges. Its training-free and few-shot design directly targets the data limitation constraint, as it can operate with only a small number of normal tote images per SKU and requires no defect examples. However, in practice, performance will strongly depend on the representativeness of these normal samples, which must capture the typical product appearances and orientations within the tote. The reliance on pre-trained foundation models yields class-agnostic generalisation, supporting scalability to thousands of SKUs without retraining.

The component-based segmentation and within-component patch comparison can potentially isolate individual products within a tote, enabling anomaly reasoning at the component level. Yet, the approach has not been explicitly evaluated in cluttered, multi-instance industrial scenes. In such conditions, segmentation quality may degrade due to occlusions, dense packing, or reflective surfaces, leading to fragmented or merged masks that propagate errors through subsequent modules.

Furthermore, UniVAD's dependence on large-scale vision models such as Grounded SAM and CLIP introduces substantial computational cost, which may limit its applicability in real-time, high-throughput inspection pipelines. These limitations could be mitigated by adopting lighter segmentation backbones, adaptive mask refinement, or targeted domain augmentations.

Overall, UniVAD represents a promising step towards universal, few-shot anomaly detection. Its unified architecture and class-agnostic logic make it conceptually well-suited for tote-level inspection, but successful deployment will rely on achieving robust segmentation consistency, efficient inference, and adequate normal sample coverage under complex, cluttered conditions.

E.2.3. AnomalyGPT

AnomalyGPT [5] frames industrial anomaly detection as a visual question–answering task powered by a large vision–language model. A frozen visual encoder is aligned with a conversational LLM and augmented with two lightweight additions that inject industrial anomaly knowledge. The first is a feature-matching decoder for pixel-level localisation. The second is a prompt learner that translates visual anomaly cues into structured prompt embeddings. Its novelty lies in coupling pixel-level feature matching with language-based reasoning, enabling unified visual and textual anomaly understanding within a single model. These embeddings allow the LLM to produce discrete anomaly decisions and coarse spatial descriptions without threshold tuning.

The overall architecture of AnomalyGPT is illustrated in Figure E.3. Given a query image, a frozen ImageBind-based vision transformer extracts a global image embedding together with patch features from four intermediate encoder stages. A lightweight decoder then produces a pixel-level anomaly map. The decoder supports two inference modes. In the unsupervised setting, stage features are projected into the text-embedding space using learned linear projections and compared with textual prototypes that represent normal and abnormal states. The resulting similarity maps from each stage are aggregated and upsampled to form a localisation map that highlights potential defect regions. In the few-shot setting, patch descriptors from one or a few normal reference images of the same SKU are stored in a memory bank. Each query patch is matched to its nearest normal patch to yield a deviation map, enabling anomaly localisation without retraining.

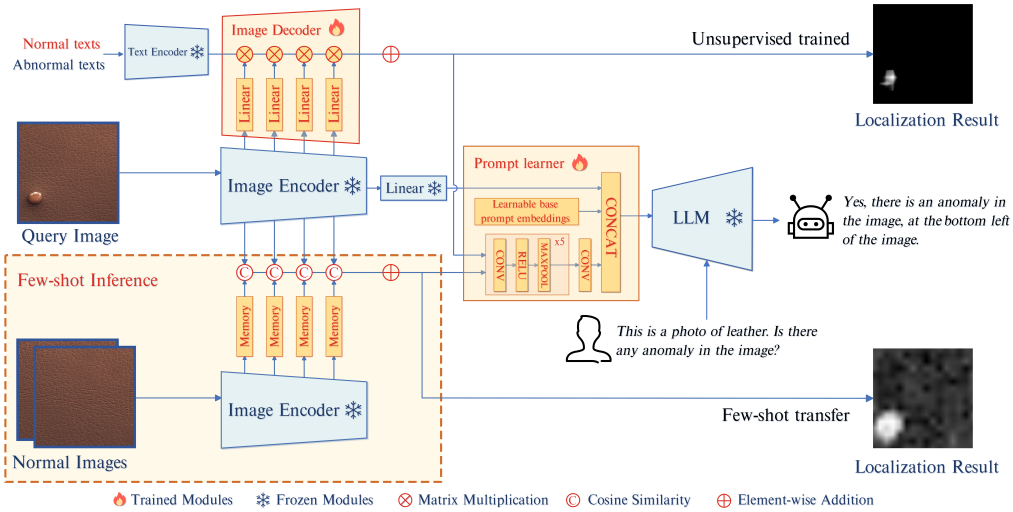


Figure E.3: AnomalyGPT architecture adapted from [5].

The prompt learner converts the decoder's localisation map into prompt embeddings using a small convolutional network and combines these with a fixed set of learnable base prompt tokens. Together with the global image embedding, the tokens are passed to the LLM (Vicuna-7B), which returns a discrete anomaly decision and a coarse spatial description on a predefined 3×3 grid. Synthetic anomaly data are generated with CutPaste and Poisson blending for training [57, 58]. The optimisation minimises a joint objective of cross-entropy on the language output and focal and dice losses on the localisation maps. Only the decoder, the prompt learner, and the linear projection layers are updated, while the image encoder and LLM remain frozen to preserve general visual–linguistic capability. To prevent catastrophic forgetting, training alternates between simulated anomaly data and general vision–language samples, preserving transferability across domains.

For multi-instance tote inspection, AnomalyGPT has several advantageous properties but also notable challenges. Its few-shot and prompt-tuned design directly addresses the data limitation constraint, as it can adapt from as few as one or two normal tote images per SKU and requires no defect annotations. The use of a frozen ImageBind encoder and language-based reasoning enables class-agnostic generalisation, allowing a single model to scale across thousands of SKUs without retraining. However, practical performance will depend on the representativeness of the available normal samples, which must capture the typical appearance, orientation, and illumination conditions of products within the tote. Insufficient variation in these references may lead to false positives when normal instances appear in unseen poses or under occlusion.

The pixel-level reasoning of the decoder and the LLM's ability to verbalise coarse anomaly locations provide a degree of robustness in cluttered, multi-instance scenes. By operating on patch-level features, the model can in principle isolate local defects on individual products. However, the method has been primarily validated on single-object images from datasets such as MVTec-AD and VisA, which contain limited clutter and occlusion. In realistic tote conditions with overlapping items or reflective packaging, the image–text alignment may degrade, leading to incomplete or imprecise localisations. Additional adaptation, such as prompt calibration or synthetic multi-object augmentation, may therefore be required to maintain reliability in dense scenes.

From a computational perspective, the frozen vision backbone and lightweight decoder allow efficient feature extraction, while the LLM introduces the main latency overhead. The reference implementation employs Vicuna-7B with concise textual responses, suggesting that inference within a few seconds per tote is achievable on a cloud GPU. However, processing high-resolution images or large memory banks may increase runtime, requiring optimisations such as mixed-precision inference or input downscaling. Overall, AnomalyGPT offers a conceptually strong foundation for threshold-free and class-agnostic anomaly detection at the tote level. Its success in deployment will rely on maintaining alignment robustness in cluttered, multi-instance settings, ensuring representative normal coverage, and achieving consistent real-time throughput across diverse SKU categories.

E.2.4. Critical Summary and Practical Implications

AnomalyDINO [4], UniVAD [19] and AnomalyGPT [5] advance data-efficient, class-agnostic detection by using strong pre-trained features and training-free or few-shot pipelines. AnomalyDINO offers efficient patch-level matching with good localisation from very few normal samples. UniVAD adds component-level reasoning that can bridge local defects and higher-level structure. AnomalyGPT contributes unified pixel and language outputs that support human-interpretable decisions.

The dominant limitation across all three methods is the need for a representative normal set per product. Normal coverage must span orientations, partial views and illumination. Maintaining such coverage for about ten thousand SKUs that change

over time is difficult. A second limitation is the gap between typical evaluation data and real tote scenes. Most results rely on single-item images with limited clutter and occlusion, which understates the difficulty of instance segmentation in dense totes. In practice, scalable deployment will require strategies that reduce normal set maintenance, such as compact and refreshable memories, aggressive pose and occlusion augmentation, and robust instance proposals before scoring.

This thesis directly addresses these gaps by developing a class-agnostic anomaly detection framework that minimises dependence on SKU-specific normal data (RQ1), ensures robust detection under clutter, occlusion, and orientation variability in multi-instance totes (RQ2), and meets real-time latency requirements for operational deployment in warehouse environments (RQ3).

E.3. Datasets and Evaluation Metrics

Evaluating SOTA methods for the tote problem requires datasets and metrics that reflect its unique challenges of limited data, cluttered multi-instance scenes, and real-time applicability. Three complementary dataset types are therefore selected: End-to-End tests combining segmentation and anomaly detection, Domain and Scoring tests focused on defect evaluation, and Proposal and Masking tests assessing instance segmentation under clutter. Correspondingly, performance is evaluated using anomaly detection, instance segmentation, and computational efficiency metrics to ensure both methodological robustness and operational suitability.

E.3.1. Benchmark Datasets

Benchmark datasets that reflect the tote problem, characterised by multi-instance and randomly oriented products requiring defect detection under occlusion, are scarce. We therefore select three complementary types. First, multi-object datasets with anomaly labels in compositionally complex scenes to assess end-to-end anomaly detection and localisation. Second, grocery and industrial datasets focused on defects, damage and freshness, where anomaly scores are verified against product-level labels in mostly single-instance views. Third, cluttered product scenes without anomaly labels to evaluate object detection and instance masking for multi-instance totes. A summary of the curated datasets is provided in Table E.1.

Table E.1: Compact overview of eight datasets aligned with the three evaluation vectors. Symbols indicate presence (✓), absence (×), or limited presence (∼) of key factors. Roles: **End** for end-to-end anomaly evaluation, **Defect** for defect or quality scoring, and **Mask** for detection or masking assessment.

Dataset	#Classes	Multi-inst.	Rand. Orient.	Pixel GT	Role	Relevance
MVTec AD [1]	15	×	×	✓	Defect	Baseline for pixel-level AD. Single-object scenes, no clutter or orientation change.
VisA [2]	12	∼	∼	✓	Defect	Industrial items with mild pose variation. Controlled AD, limited clutter.
MVTec LOCO AD [20]	5	✓	×	✓	End	Logical anomalies (missing/wrong). Tests compositional reasoning.
Kaputt [21]	48k	×	✓	×	End	Retail trays with real packaging damage. Closest match to tote setup.
ARMBench [23]	N/A	✓	✓	×	Mask	Realistic multi-object masking in cluttered bins. Defects labelled only post-pick.
FruitQ [59]	11	×	×	×	Defect	Freshness-labelled produce. Used for quality score calibration.
MVTec D2S [22]	60	✓	✓	×	Mask	Shelf scenes with heavy clutter. Evaluates instance segmentation.
Custom Tote (Ours)	10k	✓	✓	∼	End	Multi-instance stock totes with random pose, orientations, and occlusion

As shown in Table E.1, no existing dataset combines multi-instance clutter, random orientation, and pixel-level anomaly labels. MVTec AD and VisA remain industrial baselines, though both contain mostly single-object scenes with fixed orientations. FruitQ plays a similar role for freshness scoring, but lacks multi-instance variation. MVTec D2S and ARMBench are useful for assessing object masking and segmentation under clutter before anomaly detection is applied. ARMBench additionally includes a robot-induced defect subset, though these labels are captured post-pick rather than within cluttered scenes.

Among the end-to-end datasets, MVTec LOCO AD focusses on logical and structural anomalies but lacks clutter and random orientation. Kaputt best reflects tote conditions with retail logistics imagery and random poses, yet omits multi-instance clutter and pixel-level annotations. The Custom Tote dataset fills this gap by combining multi-instance clutter, random orientation, and real-world tote imagery. With the potential addition of pixel-level ground truth, it provides a representative benchmark for anomaly detection in operational logistics environments.

E.3.2. Evaluation Metrics

To assess the performance of an anomaly detection pipeline, evaluation is conducted along three dimensions that reflect its performance, anomaly scoring and localisation accuracy, instance- and mask-level consistency, and computational efficiency. Together, these dimensions capture both methodological robustness and operational applicability in large-scale fulfilment environments.

Anomaly scoring and localisation metrics.

The performance of anomaly detection models is typically evaluated using threshold-independent metrics that assess both discriminative and localisation capability. The Area Under the Receiver Operating Characteristic (AUROC) remains the most widely used measure of discriminative performance. It quantifies how effectively a model distinguishes normal from defective samples across all thresholds by integrating the true-positive rate over the false-positive rate [60]. When computed at image level, AUROC

reflects a model's ability to identify whether a tote contains any defect. At pixel level, the same principle is extended to each pixel or mask, yielding the Pixel-level AUROC or its region-based variant, the Area Under the Per-Region Overlap (AUPRO) [1]. These localisation metrics evaluate how accurately the predicted anomaly map aligns with the ground-truth defect regions. For imbalanced datasets, threshold-dependent measures such as Precision, Recall, and their combined F1-score are also reported to capture trade-offs between false positives and missed detections. These evaluation criteria are most relevant for the defect and quality scoring datasets, including MVTec AD, VisA, and FruitQ, where explicit anomaly labels are available.

Instance- and mask-level evaluation metrics.

In multi-instance tote inspection, segmentation accuracy is critical to ensure that each product is correctly isolated before defect scoring. The most widely used metric for evaluating such instance segmentation performance is the mean Average Precision (mAP), which measures the area under the precision–recall curve across Intersection-over-Union (IoU) thresholds. Variants such as mAP_{50} and mAP_{75}, computed at IoU thresholds of 0.5 and 0.75 respectively, provide lenient and strict measures of spatial alignment between predicted and ground-truth masks. When averaged over multiple IoUs (0.5–0.95), mAP rewards models that achieve consistent mask accuracy across varying overlap criteria. These metrics are standard in dense multi-instance segmentation benchmarks such as MVTec D2S [22] and ARMBench [23], where high object counts, occlusion, and pose variation challenge precise segmentation. For this work, high mAP values indicate robust mask generation under cluttered tote conditions, ensuring reliable inputs for subsequent anomaly detection stages.

Computational efficiency metrics.

In large-scale fulfilment settings, anomaly detection models must meet strict latency and resource constraints while remaining scalable across thousands of SKUs. The most direct measure of efficiency is the inference time per image, or inversely, the frames per second (FPS), typically reported under specified hardware configurations (CPU or GPU). Lightweight architectures such as EfficientAD achieve millisecond-level inference on high-end GPUs [17], whereas foundation-based models like UniVAD and AnomalyGPT incur higher runtimes in exchange for broader generalisation and reasoning capability [5, 19].

Complementary indicators such as model size, parameter count, and floating-point operations (GFLOPs) quantify computational complexity and hardware scalability. Data efficiency is further assessed through the number of normal images required per SKU and the associated training time, which directly affect deployability at scale. Together, these measures characterise the trade-off between accuracy, latency, and scalability that defines the practical viability of training-free anomaly detection in logistics environments.

E.3.3. Summary

In summary, this section shows that no single dataset or metric can fully capture the complexity of the tote problem. A comprehensive evaluation therefore requires a combination of dataset types that together assess object segmentation in cluttered scenes and fine-grained anomaly detection. Performance must be measured through anomaly scoring and localisation accuracy, segmentation quality, and computational efficiency to ensure suitability for deployment in automated fulfilment environments.

E.4. Identified Knowledge Gap

Recent state-of-the-art methods leverage powerful pre-trained models to achieve data-efficient and class-agnostic anomaly detection. Patch-based and memory-driven matching with strong backbones delivers accurate localisation from very few references [3, 4, 13]. Unified and vision–language pipelines extend this with component reasoning and textual cues that improve generalisation without per-SKU training [5, 12, 19]. These directions set a strong foundation for automated quality inspection.

Despite this progress, a practical gap remains for the tote problem. First, data limitation and scalability are unresolved. All three SOTA methods rely on a representative normal reference for each product that covers pose, partial views and illumination. In practice, zero to a few normal tote images are available per SKU, possibly complemented by a single supply-chain image at inference. Maintaining adequate normal coverage for roughly ten thousand changing SKUs is a data logistics problem that current methods and evaluations do not address. Non-parametric memories and nearest neighbour scoring used by SOTA methods are sensitive to incomplete normal sets and to shifts in normal appearance [4, 5, 19].

Multi-instance clutter and occlusion present a second unresolved challenge. A large share of evaluations for recent methods is conducted on single-instance, fixed orientation benchmarks such as MVTec AD [1]. Even when multi-view or category-diverse datasets are used, scenes often lack the heavy overlap, pose variability and partial visibility that characterise real totes. Under these conditions, class-agnostic detectors risk false positive when encountering unseen but normal shape deformations, orientations or occlusions that are common in packed totes. Current evaluations provide limited evidence that these methods can maintain robustness across different instance counts, overlap patterns and pose variability.

Third, real-time performance under operational latency remains insufficiently validated for foundation-heavy pipelines. Large vision–language backbones and segmentation assistants such as CLIP, RAM and Grounded SAM introduce substantial computational and memory cost [5, 12, 19, 55, 56]. Achieving real-time feasibility in an automated warehouse therefore requires a careful balance between accuracy and efficiency, supported by targeted optimisation of model size, inference throughput and system integration.

When examining existing benchmarks, similar limitations become evident. No single public dataset combines multi-instance clutter, high occlusion, grocery logistics imagery and anomaly annotations. Industrial anomaly detection datasets such as MVTec AD, MVTec LOCO and VisA lack the cluttered, multi-object compositions and orientation diversity characteristic of tote scenes [1, 2, 20]. More recent datasets such as KAPUTT extend anomaly diversity and orientation variability but still focus on single-object views [21]. Conversely, proposal and masking datasets such as D2S and ARMbench include cluttered layouts but omit explicit anomaly labels [22, 23]. This lack of integrated benchmarks restricts comprehensive and comparable evaluation and increases the risk of overfitting methods to simplified, single-object scenarios.

This thesis therefore aims to address these gaps by developing a multi-instance robust anomaly detection pipeline that remains effective under clutter, orientation variability, and occlusion. The approach seeks to reduce dependency on large per-SKU normal sets to address the data limitation and scalability challenges, while maintaining accuracy within real-time operational constraints. The intended contribution is a practical, hybrid framework that integrates instance-aware representation, adaptive reference management and compute-efficient inference. Together, these components aim to reduce false positives from normal variation while preserving sensitivity to genuine product and packaging defects. The proposed model will be evaluated on both established datasets and a custom benchmark that more accurately reflects the visual and operational conditions of automated warehouse environments.

F Project Plan

This chapter outlines the practical execution of the research, translating the objectives defined in the previous chapters into a structured plan. It introduces the methodological framework used to design, implement, and evaluate the tote anomaly detection pipeline, followed by the detailed planning of project activities and milestones. Together these sections describe how the proposed approach will be developed, validated, and delivered within the available timeframe and resources.

F.1. Methodology

This section presents the methodological framework for addressing the tote anomaly detection problem. It outlines the development of a class-agnostic pipeline for detecting product and packaging defects in cluttered tote images with limited reference data. The methodology covers three parts: the data strategy and baselines, the proposed detection pipeline, and the evaluation framework for assessing accuracy, localisation, and computational efficiency. Together these components form the foundation for a scalable and real-time visual inspection system for fulfilment operations.

F.1.1. Data Strategy and Baselines

The data strategy evaluates performance against the core tote challenges by combining public benchmarks with a custom dataset that reflects the operational environment. For anomaly detection, MVTec AD and VisA establish few-shot, pixel-level baselines under controlled conditions [1, 2]. Kaputt adds realistic packaging defects and pose diversity [21], while MVTec LOCO AD extends evaluation to multi-item layouts [20]. FruitQ complements these by testing generalisation to freshness [59]. For segmentation robustness in cluttered scenes, MVTec D2S and ARMBench quantify instance recall and resistance to occlusion [22, 23].

A custom tote dataset will be developed from FC imagery to reproduce the camera geometry, background variation, packaging materials, and lighting found present in deployment. The dataset will contain labelled anomalies (image- or pixel-level) and will exclude any overlap with the reference base. It provides an end-to-end benchmark under realistic multi-instance clutter, occlusion, and pose variation.

The state-of-the-art (SOTA) methods identified in the literature are used as baselines to benchmark the proposed framework. AnomalyDINO implements training-free patch-based matching with strong foundation features [4]. UniVAD introduces a hybrid framework with component-aware and logical reasoning [19]. AnomalyGPT integrates a vision–language interface that aligns pixel-level cues with semantic prompts [5]. All baselines are evaluated using identical preprocessing, scoring, and runtime conditions to ensure comparability.

To address limited normal data per SKU, a minimal and scalable reference base will be created. Each SKU will have up to three high-quality "normal" tote images selected to capture variation in instance count, pose, and orientation. Where available, auxiliary imagery from supply-chain databases and customer-facing app catalogues will augment this base to improve appearance diversity. These references will be programmatically structured, potentially supported by large language models to maintain balanced coverage across products. This compact, structured reference base is designed for scalability to approximately 10,000 SKUs and adaptability to dynamic product assortments.

F.1.2. Conceptual Model

The proposed pipeline follows the flow illustrated in Figure F.1. During inference, the system retrieves only the reference base of the active SKU, which is constructed as described in subsection F.1.1. To reduce latency, the reference base may store precomputed patch features rather than full images. A frozen image encoder then extracts patch embeddings from both the query image and the references. Different backbones will be evaluated, including CLIP and DINO variants, to compare their feature quality and computational efficiency.

Component segmentation is optional and evaluated for its trade-off between accuracy and runtime. Option A applies RAM with Grounded SAM, following UniVAD's component discovery stage [19], while Option B uses zero-shot foreground masking as in AnomalyDINO [4]. Masks are refined with feature-space K-means to remove noise and improve instance consistency under clutter and occlusion.

Patch features are then compared only within component regions. The matching uses nearest-neighbour distances between query patches and the reference patch bank, a PatchCore-style mechanism adopted by AnomalyDINO and applied in UniVAD's component-aware matching [3, 4, 19]. Restricting the comparison to component masks reduces background false positives and ensures that scoring focuses on the products rather than tote structure.

Text guidance is optional. When enabled, short prompts describing normal and defect states are embedded in the same feature space and compared with patch features. This follows the semantic scoring strategy used in UniVAD and the vision–language pathway introduced by AnomalyGPT [5, 19]. The outputs from component-level patch matching and text-based scoring are fused into a single anomaly map.

The decision stage operates on this map. A basic configuration applies a threshold to the average of the most anomalous patches to produce a tote verdict. Alternatively, a lightweight LLM head can interpret the heatmap and image context to return a concise

judgement with rationale, as proposed in AnomalyGPT [5]. This approach maintains scalability while providing interpretability where required.

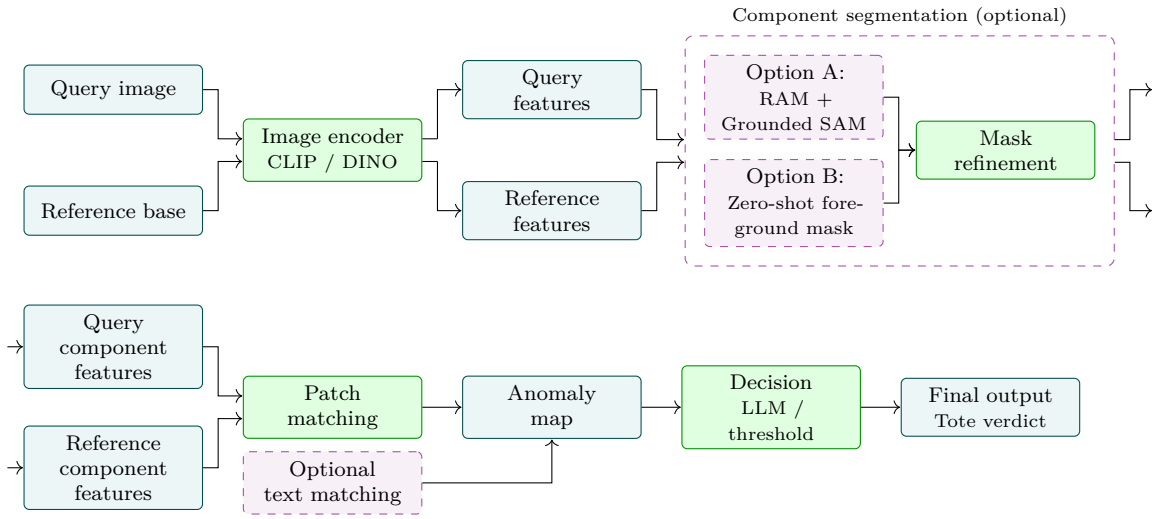


Figure F.1: Conceptual pipeline for tote anomaly detection. The model compares query and reference images using patch-level features from pre-trained encoders. Optional segmentation and text guidance modules refine localisation and interpretability, producing an anomaly map and final tote-level decision.

F.1.3. Evaluation Framework

Evaluation follows three dimensions defined in the literature review, namely anomaly scoring and localisation accuracy, instance and mask consistency, and computational efficiency. Public datasets benchmark sub-tasks, while the custom tote dataset provides end-to-end evaluation. All models use identical preprocessing, input resolution, and masking options, and results are reported as the mean and standard deviation across runs.

Anomaly scoring and localisation are assessed using the Area Under the Receiver Operating Characteristic (AUROC) at both image and pixel level. The pixel-level AUROC and the Area Under the Per-Region Overlap (AUPRO) quantify the alignment between predicted anomaly maps and ground-truth defect regions. For imbalanced subsets, Precision, Recall, and the F1-score are also reported at an operating threshold selected based on the validation results to balance false positives and missed detections.

Instance and mask quality are evaluated using mean Average Precision (mAP) at Intersection-over-Union (IoU) thresholds of 0.50, 0.75, and the averaged range from 0.50 to 0.95. Instance recall at fixed IoU thresholds is included to highlight segmentation robustness under heavy occlusion. These metrics are computed on MVTec D2S, ARMBench.

Computational efficiency and scalability are characterised by the inference latency per tote image, throughput in frames per second (FPS), model size in megabytes, and the number of floating-point operations (GFLOPs) per forward pass. Data efficiency is tracked as the number of normal images required per SKU within the reference bank. Two runtime profiles are compared, a fast path using zero-shot foreground masking, and an accurate path using RAM with Grounded SAM proposals.

Experiments are conducted locally, with memory, batch size, and input resolution reported. Real-time inference performance is evaluated on a managed cloud GPU service using the same configuration intended for deployment.

F.2. Planning

This section outlines the overall planning of the thesis project, structured into four main phases. Each phase consists of several work packages (WP) with defined objectives, time allocations, and deliverables. The total project duration is nine months, spanning from September 2025 to May 2026. An overview of key milestones and dates is presented in Table F.1, while the detailed project planning and timeline are illustrated in Figure F.2.

Phase 1: Literature Review and Research Proposal (8 weeks)

This phase establishes the theoretical and methodological foundation of the project. It involves completing the literature review, defining the research objective and questions, and identifying the core challenges related to data scarcity, multi-instance clutter, and real-time constraints. The outcome is a finalised research proposal that outlines the experimental scope, selected datasets, baseline methods, and evaluation criteria. The phase concludes with the submission of the research proposal deliverable.

WP 1: Scope, Literature Review, and Problem Definition (6 weeks)
Define the thesis scope, review relevant literature, identify limitations of current methods, and refine the research problem. *Deliverable: Comprehensive literature study and clearly defined research gap.*

WP 2: Research Objective and Proposal Development (2 weeks)
Formulate the research objective and questions based on the identified gap. Define the experimental setup, including dataset selection, baseline methods, and evaluation metrics. *Deliverable: Completed research proposal and project plan.*

Phase 2: Model Development (11 weeks)

This phase focuses on implementing and developing the proposed anomaly detection framework. The work includes experimenting with state-of-the-art methods, creating a custom dataset, and developing the complete processing pipeline for anomaly detection in cluttered tote scenes. It will also explore data management and augmentation strategies to address the limited availability of normal samples. The phase concludes with a working prototype capable of detecting anomalies in tote images and the midterm review deliverable.

WP 3: Data Collection and Preparation (2 weeks)
Collect and organise normal tote imagery for the reference base. Create a custom dataset for testing and evaluation. *Deliverable: Reference base and custom tote dataset.*

WP 4: Baseline Model Experimentation (4 weeks)
Implement and experiment with state-of-the-art anomaly detection methods to identify suitable components and determine which can be reused or improved. *Deliverable: Baseline evaluation and selected framework components.*

WP 5: Framework Design and Integration (5 weeks)
Develop the proposed anomaly detection framework by integrating and extending baseline components. Focus on scalable reference handling, few-shot adaptability, and robust feature reasoning under occlusion. *Deliverable: First functional prototype.*

Phase 3: Evaluation and Analysis (10 weeks)

In this phase, the developed model is evaluated, validated, and refined to address challenges such as changing product catalogues, computational efficiency, and robustness under real fulfilment conditions. The framework's performance will be compared against baseline methods using both established benchmarks and the custom dataset. The outcome of this phase includes a validated model and a complete draft of the thesis report.

WP 6: Experimental Evaluation and Benchmarking (3 weeks)
Evaluate the developed framework against baseline models using established benchmarks and the custom tote dataset. Measure detection accuracy, localisation quality, and runtime efficiency. *Deliverable: Quantitative and qualitative evaluation results.*

WP 7: Ablation and Sensitivity Analysis (2 weeks)
Analyse the influence of key design choices such as segmentation strategy, feature extraction, memory management, and data augmentation. *Deliverable: Ablation study and performance analysis report.*

WP 8: Proof of Concept: Drone-Based Validation (2 weeks)
Test the framework on imagery collected using a drone to assess robustness to motion, perspective, and video-based inputs. *Deliverable: Proof-of-concept validation report.*

WP 9: Final Model Validation and Thesis Drafting (3 weeks)
Refine and finalise the model based on experimental outcomes. Integrate the final results, discussion, and analysis into the thesis document. *Deliverable: Validated model and full thesis draft.*

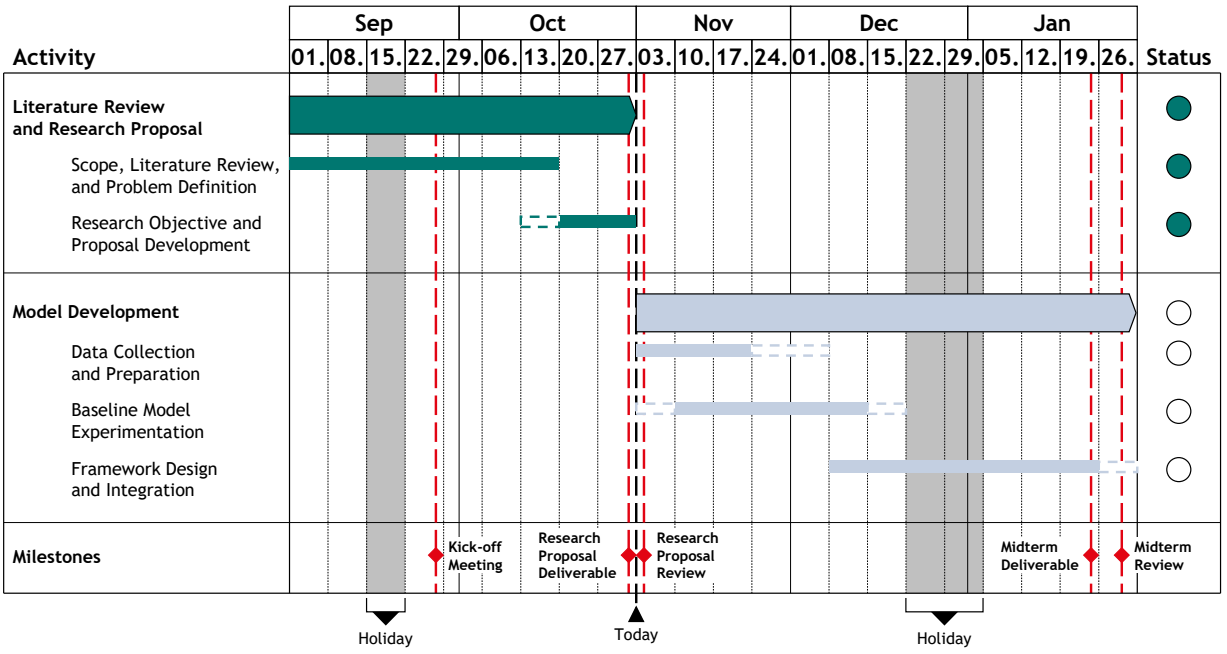
Phase 4: Research Dissemination (7 weeks)

The final phase focuses on consolidating and presenting the research outcomes. It includes finalising the written thesis, preparing visual materials, and completing the green-light review and defence preparation. The phase concludes with the submission of the final thesis and the formal defence.

WP 10: Documentation, Presentation, and Defence Preparation (7 weeks)
Finalise the written thesis, prepare visual materials and presentation slides, and complete the green-light review and defence preparation. *Deliverable: Submitted final thesis and defence presentation.*

Table F.1: Proposed thesis timeline with key milestones.

Week	Milestone	Proposed Date
3	**Holiday**	**September 15 2025 – September 21 2025**
4	Kick-off Meeting	September 26 2025
9	Research Proposal Deliverable	October 31 2025
10	Research Proposal Review	November 4 2025
17–18	**Holidays**	**December 22 2025 – January 4 2026**
21	Midterm Deliverable	January 23 2026
22	Midterm Review	January 29 2026
32	Submit Draft Thesis	April 10 2026
34	Green Light Review	April 24 2026
35	Request Examination	April 30 2026
37	Research Portfolio Submission	May 15 2026
39	Thesis Defence	May 28 2026



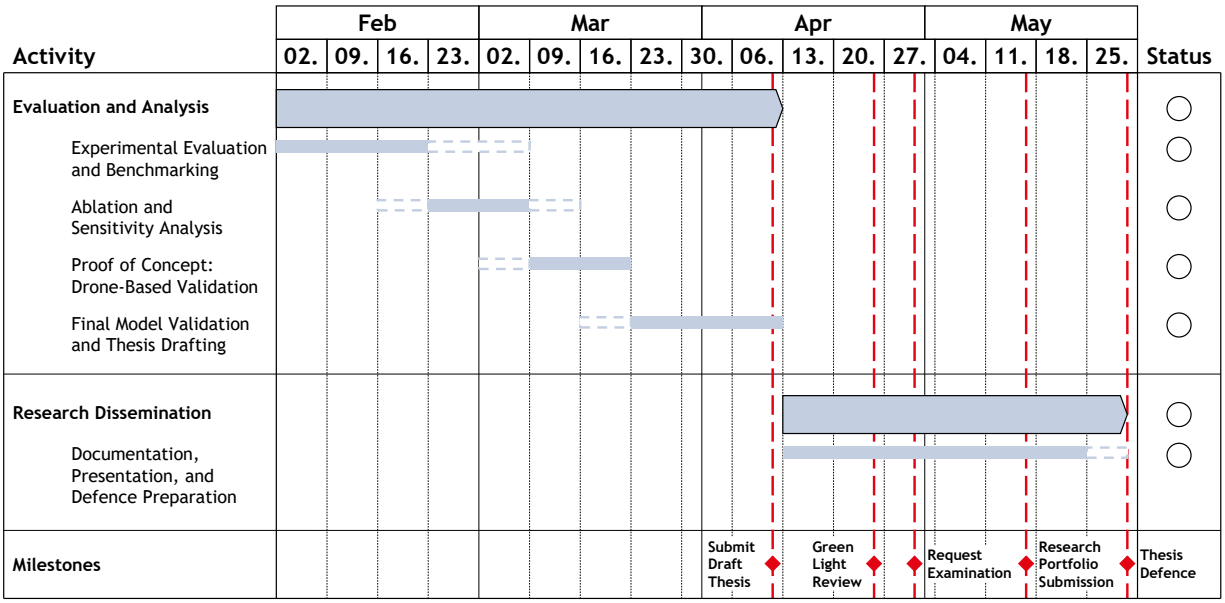


Figure F.2: Proposed project planning and timeline developed with Think-cell. The plan is split into two parts for readability: September–January (top) and February–May (bottom).