Revista de Informática Teórica e Aplicada - RITA - ISSN 2175-2745
Vol. 33, Num. 2 (2026) 115-122

RESEARCH ARTICLE

# Two-Stage Fine-Tuning of Object Detectors for Industrial Environments

## Ajuste Fino em Duas Etapas de Detectores de Objetos para Ambientes Industriais

Gabriel Moreira[1]*, Ian Otoni[1], Michel M. Silva[1], Thiago L. Gomes[1]

**Abstract:** Automated quality control in industrial production has the potential to reduce errors and provide real-time information. However, the inspection of secondary packaging, such as counting boxes in crates, still represents a challenge, since manual methods are slow and error-prone, while automatic methods are limited by the scarcity of domain-specific datasets and the high cost of annotation. This paper proposes an efficient and low-cost two-stage object detection workflow for automatic box counting. The central novelty lies in the integration of the **Segment Anything Model** (SAM) to accelerate the creation of a high-quality dataset from production line videos, making model specialization economically feasible. Initially, a generalist model is trained on heterogeneous datasets to capture general visual features. Then, the model is fine-tuned with the domain-specific dataset of approximately 750 images. Experiments with YOLOv11x and Faster R-CNN achieved mAP@0.5 above 98%, with YOLOv11x showing higher accuracy and faster inference. These results demonstrate the efficiency of the proposed approach, establishing it as a replicable and low-cost solution for monitoring secondary packaging in industrial environments.
**Keywords:** industrial object detection — automated quality control — two-stage fine-tuning — computer vision

**Resumo:** O controle de qualidade automatizado na produção industrial tem potencial para reduzir erros e fornecer informações em tempo real. No entanto, a inspeção de embalagens secundárias, como a contagem de caixas em engradados, ainda representa um desafio, pois métodos manuais são lentos e suscetíveis a falhas, e métodos automáticos são limitados pela escassez de datasets de domínio especifico e pelo alto custo de anotação. Este trabalho propõe um fluxo de trabalho eficiente e barato de detecção de objetos em duas etapas para a contagem automática de caixas. A novidade central reside na integração do **Segment Anything Model** (SAM) para acelerar a criação de um dataset de alta qualidade, a partir de vídeos da linha de produção, tornando a especialização do modelo economicamente acessível. Inicialmente, um modelo generalista é treinado em conjuntos de dados heterogêneos para capturar características visuais gerais. Em seguida, o modelo é refinado com o conjunto de dados específico do domínio de aproximadamente 750 imagens. Experimentos com YOLOv11x e Faster R-CNN resultaram em mAP@0,5 superior a 98%, com YOLOv11x apresentando maior precisão e velocidade de inferência. Esses números demonstram a eficiência da abordagem proposta, consolidando-a como uma solução replicável e de baixo custo para o monitoramento de embalagens secundárias em ambientes industriais.
**Palavras-Chave:** detecção de objetos industriais — controle de qualidade automatizado — ajuste fino em duas etapas — visão computacional

[1] *Departamento de Informática, Universidade Federal de Viçosa (UFV), Brazil*
***Corresponding author**: gabriel.m.marques@ufv.br
**DOI:** http://dx.doi.org/10.22456/2175-2745.150904 • **Received:** 14/10/2025 • **Accepted:** 03/11/2025


## 1. Introduction

Among the technological advances of recent years, the integration of advanced digital technologies into production processes has given rise to yet another industrial era [1]. In this evolutionary process, progress in research on machine learning, particularly in computer vision, has proven effective across various industrial sectors. For example, Zribi *et al.* [2] demonstrate that the application of image classification to plastic consumables in laboratories can partially replace manual inspection, increasing accuracy and reducing costs. Similarly, Khan *et al.* [3] present a framework that combines camera and sensor data with convolutional neural networks to create

digital twins of factories, enabling object tracking and KPI extraction in complex industrial environments. This implies that computer vision can automate repetitive tasks, reduce human errors, and provide real-time insights, highlighting its transformative potential in production processes.

In the industrial sector, quality control stands out as one of the most demanding and critical activities. Its implementation is intended to guarantee that final products conform to predefined standards, thereby reducing the incidence of defects and fostering higher levels of operational efficiency. For this reason, quality control has become an essential component of most automated production lines [4]. Despite its importance, many quality control processes still do not take full advantage of recent advances in machine learning, particularly in computer vision, due primarily to the limited availability of datasets and the absence of robust, task-specific models.

In this context, a particularly relevant challenge concerns verifying the correct quantity of items in secondary packaging, such as boxes or crates. The large scale of contemporary production makes traditional approaches, including manual inspection, both inefficient and highly susceptible to human error. These limitations underscore the importance of adopting automated monitoring systems based on computer vision and machine learning, which can provide accurate, scalable, and real-time solutions. Nevertheless, despite their potential, existing applications in this domain remain limited, as few studies address secondary packaging inspection, and publicly available datasets or tailored models are still scarce.

To address the scarcity of tailored datasets, a pragmatic workflow is required to bridge the gap between general-purpose models and specific industrial needs. The primary bottleneck is often the prohibitive cost and time associated with manually annotating data for the production environment. This challenge causes even robust pre-trained models to fail to achieve optimal performance without specialized fine-tuning. Motivated by this, this paper proposes and validates a two-stage fine-tuning pipeline for automated counting of boxes in crates. The first stage involves training a generalizable model on broad, heterogeneous datasets to learn visual features of the target object - the juice boxes. Subsequently, the model is fine-tuned on a high-quality, domain-specific dataset consisting of approximately 750 images extracted from videos recorded directly on the production line.

Although modest in size, the created dataset was sufficient to achieve high performance since the model had already learned general features from the previous stage and its a domain specific task. Furthermore, the creation of this dataset made economically viable by using the Segment Anything Model (SAM) [5] for semi-automated annotation, that substantially reduces manual labeling effort. Also, collecting data directly from the production line exposes the model to realistic variations in box positioning, lighting and crate configurations, improving robustness without large operational investments.

In our experiments, we trained two state-of-the-art models, YOLOv11x [6] and Faster R-CNN [7], following the proposed methodology. The results demonstrate that both models achieved outstanding performance in detecting packages within crates, thereby validating the effectiveness of the two-phase training strategy. With mAP@0.5 metrics superior to 98% for both architectures, the models exhibited a strong ability to accurately localize the objects of interest. These findings highlight the potential of the proposed approach to yield specialized models adapted to real operating environments, enhancing robustness and ensuring greater reliability in automated inspection processes.

## 2. Related Work

In this section, we highlight the datasets that enable the development of this and other works related to the problem of package counting in crates, as well as studies that integrate computer vision techniques into the processes of industrial production lines.

### 2.1 Datasets Applied to the Problem of Crate Counting

When working in the field of machine learning, it is common in many research endeavors to encounter scenarios where only a limited amount of labeled data is available, this was indeed the case for the problem of package counting in industrial environments. In this context, platforms such as Roboflow [8] provide an alternative means of addressing the issue of data scarcity in specific domains, as they allow users around the world to contribute to various projects by making their own datasets publicly available. However, since it is a collaborative platform, it is natural that not all of the datasets provided present the same level of quality and standardization found in more consolidated databases.

To address this challenge, and with a focus on object detection for counting juice boxes in crates, we collected several datasets that present generalist characteristics of packaging in general. The datasets [9, 10] contain images of dairy products, aimed at identifying this specific type of packaging. The datasets [11, 12] primarily focus on juice box packaging, while [13] broadens this scope by including, in addition to juice boxes, soda cans. Finally, [14] gathers images of various products and packaging, offering a more diverse scenario that combines juice boxes and dairy product packaging.

Although we did not find consolidated datasets specifically addressing the problem of counting juice packages in crates, it is relevant to highlight related initiatives that deal with similar contexts. Yang et al. [15] provide a dataset composed of over 16,000 images and approximately 250,000 instance annotations of stacked boxes and bales in industrial scenarios, aimed at object detection and semantic segmentation tasks. Despite the fact that it represents an application domain close to that of this work, most of the images were not captured from a top-down perspective, which limits their suitability for our context, as the models proposed here are applied directly to

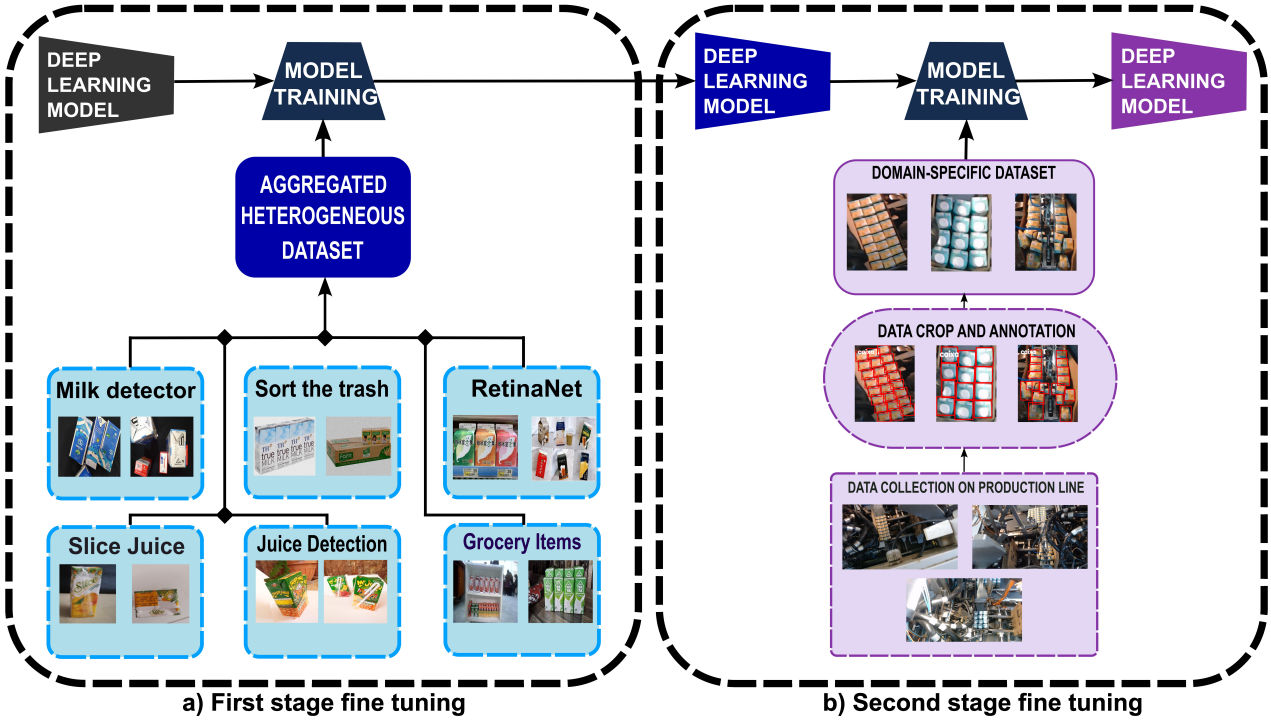


**Figure 1.** Overview of the proposed methodology for automated box counting. Stage 1 involves training a generalist object detection model on aggregated heterogeneous datasets, while Stage 2 fine-tunes the model using a domain-specific dataset collected from the production line to improve robustness and reliability in real-world conditions.

production lines. Furthermore, our focus is on counting items within the packages, rather than specifically on the bales.

### 2.2 Computer Vision in Industrial Production Processes

The adoption of technology within production cycles has driven research that not only automates but also optimizes industrial operations. In this scenario, computer vision plays a strategic role, as it is capable of extracting relevant information from images and videos, supporting monitoring and, consequently, the quality control of production lines.

Rahman *et al.*[4] highlight this premise by stating that it is impractical to continue relying on traditional monitoring carried out by humans, as it presents several limitations that compromise quality control in large-scale production chains. The authors review image-processing-based defect detection methods applied to the beverage industry, demonstrating the effectiveness of these techniques in identifying major issues such as shape, color, and fill-level defects.

As mentioned earlier, Zribi *et al.*[2] propose an automated monitoring system for plastic consumables used in analytical laboratories, where a deep learning model based on computer vision is employed to categorize different images of vials that may be either empty or filled with an anticoagulant. The results indicate that, in terms of accuracy, the proposal is competitive with traditional models while requiring significantly fewer resources.

In this regard, more recent works, such as that of Khan *et al.*[3], extend the scope of computer vision beyond quality inspection, exploring its integration with digital twins in industrial production lines. The proposed approach combines camera and sensor data with advanced 3D reconstruction techniques and convolutional neural networks to enable real-time monitoring, object tracking, and the extraction of performance indicators such as availability, efficiency, and Overall Equipment Effectiveness (OEE). This approach highlights the potential of computer vision not only as an inspection tool but also as a support mechanism for strategic decision-making in smart manufacturing environments.

## 3. Methods

In this section, we present the methodological approach employed to train a neural network for object detection, aimed at supporting the automated counting of boxes in crates. The proposed methodology is structured in two main stages, as illustrated in Figure 1. In the first stage, heterogeneous datasets were aggregated to capture the general visual characteristics of the target object, the juice boxes, enabling the development of a generalizable model. In the second stage, a domain-specific dataset was constructed from videos recorded directly on the production line of a company facing this problem, thereby reflecting real-world packaging conditions and allowing the fine-tuning of a specialized model to enhance robustness and ensure greater reliability in automated inspection processes.

### 3.1 Stage 1: General Model Training

To train a general object detection model for automated counting of boxes in crates, the problem was approached in two main steps: (i) constructing a representative dataset and (ii) defining the deep learning environment and model architecture.

#### 3.1.1 Dataset Aggregation and Preparation

Given the specific nature of detecting packages within crates, selecting an appropriate dataset represented a major challenge. To address this, the first step of the methodology involved constructing a generic and comprehensive dataset. After exploring public repositories, six datasets were selected from the Roboflow [8] platform, encompassing juice, milk, and related product boxes.

- milk-detector-v2 Computer Vision Project [9] - Contains 156 images of dairy products, such as condensed milk and milk cartons, annotated with bounding boxes.
- sort the trash V4 Computer Vision Project [10] - Provides 3,798 images across various classes, including juice cartons, milk cartons, bottles, and cans, featuring polygonal annotations.
- RetinaNet Computer Vision Project [14] - Comprises 1,681 images exclusively featuring Tetra Pak-style packaging for milk, chocolate milk, and juice, all annotated with bounding boxes.
- Slice_Juice Computer Vision Project [11] - A focused dataset with 339 images, all depicting mango juice cartons annotated with bounding boxes.
- juice-detection-4-items Computer Vision Project [12] - Includes 234 images with two distinct classes, 'mr-juicy-pineapple' and 'mr-juicy-strawberry-banana', annotated using polygons.
- Grocery Items Computer Vision _Dataset [13] - A large-scale dataset with 5,359 images of milk and juice packages on supermarket shelves, annotated with polygons.

The aggregated dataset, totaling approximately 11 thousand images, presented two main challenges: i - inconsistency in annotations and ii - lack of representativeness of the target context. Firstly, the annotations were heterogeneous, with varied formats (polygons, corner coordinates) and quality issues, such as null dimensions. To overcome such problems, a pre-processing pipeline was implemented. In it, all markings were converted to the normalized scheme $(x_{\text{center}}, y_{\text{center}}, \text{width}, \text{height})$ and the original classes were unified into a single target class "box". Samples with empty annotations or even with zeroed height or width were discarded, ensuring the integrity of the final set.

Although the images portrayed the objects of interest, the scenarios were diverse (tables, floors, shelves) and did not correspond to the industrial environment with crates, where the packages tend to be adjacent to one another. Despite this contextual limitation, the objective of this dataset was to serve as a robust knowledge base, generalizing the models' ability to detect the "box" object under different lighting, angle, and partial occlusion conditions. Finally, the dataset was divided into 80% for training, 10% for validation, and 10% for testing.

**Table 1.** Hyperparameters used in the first training phase for Faster R-CNN and YOLOv11x.

| Parameter | Faster R-CNN | YOLOv11x |
|---|---|---|
| Initial Weights | COCO weights | COCO weights |
| Optimizer | SGD | AdamW |
| Learning Rate | $1 \times 10^{-4}$ | $1 \times 10^{-4}$ |
| Batch Size | 4 | 8 |
| Epochs | 40 | 50 |
| Model Size | $\sim$ 158,04 MB | $\sim$ 114,4 MB |

#### 3.1.2 General Model Training with the Aggregated Dataset

In the first training phase, we tested two model options for box detection using the aggregated generic dataset: Faster R-CNN and YOLOv11x. Faster R-CNN was implemented in PyTorch, and YOLOv11x was trained using the official Ultralytics API. Both architectures were initialized with weights pre-trained on the COCO dataset.

The principal hyperparameters employed for each model are comprehensively presented in Table 1. For Faster R-CNN, the training protocol comprised 40 epochs with a fixed learning rate of $1 \times 10^{-4}$ and a batch size of 4. In contrast, YOLOv11x was trained for 50 epochs, initialized with a learning rate of $1 \times 10^{-2}$ that underwent linear decay to $1 \times 10^{-4}$ throughout the training duration, using a batch size of 8. This scheduling strategy for YOLOv11x facilitated complete model adaptation while mitigating potential instability during later training phases. Optimization algorithms were selected following established best practices for each respective architecture: stochastic gradient descent (SGD) with momentum was employed for Faster R-CNN, while AdamW was utilized for YOLOv11x, both configurations promoting stable convergence characteristics. All network layers for both models remained unfrozen during the training process to enable comprehensive parameter optimization.

Testing these two model options allowed us to compare their performance and select the most suitable architecture for subsequent domain-specific fine-tuning, ensuring robust detection of boxes in the target production environment.

### 3.2 Stage 2: Domain-Specific Fine-Tuning

The second stage of the methodology focuses on adapting the generalist model to the target production environment. This stage is divided into two sequential steps: first, the construction of a domain-specific dataset from videos recorded directly on the production line, capturing the real-world variability of packaging conditions; second, the fine-tuning of the pre-trained generalist model on this dataset, enabling the

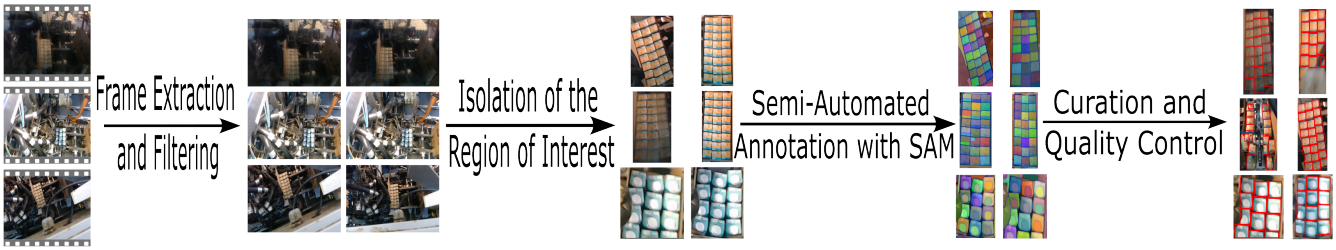


**Figure 2.** Overview of the domain-specific dataset construction pipeline. Video frames from the production line are extracted, filtered, and cropped to isolate the crate region. The Segment Anything Model (SAM) is applied for semi-automated annotation, followed by manual curation and quality control to produce the final dataset.

network to specialize for the operational context and improve robustness and reliability in automated box counting and inspection tasks.

#### 3.2.1 Domain-Specific Dataset

The domain-specific dataset was created following a multi-step pipeline based on video recordings of the production environment. Eleven videos, ranging from 25 to 100 seconds, captured a robotic arm placing packages into crates. The overall process is illustrated in Figure 2 and can be summarized in the following steps:

1. **Frame Extraction and Filtering:** Frames were extracted from all videos, yielding over 8,000 images. Manual screening was applied to select only images with clear visibility of the crate and packages, discarding frames with significant occlusion from the robotic arm or during crate changes. Approximately 1,000 candidate images were retained.

2. **Isolation of the Region of Interest (ROI):** Each selected image was cropped to isolate the crate region, removing irrelevant background information.

3. **Semi-Automated Annotation with SAM:** The Segment Anything Model (SAM) [5] was used to generate segmentation masks for all potential objects in the cropped images, accelerating the annotation process.

4. **Curation and Quality Control:** The masks produced by SAM were manually verified to retain only those corresponding to the target packages, discarding incorrect detections such as crate parts or wires. Images containing more than one annotation error, defined as either a missed package or a mislabeled object, were removed.

This rigorous process resulted in a final dataset of 752 images, specifically tailored for box detection within crates.

#### 3.2.2 Domain-Specific Fine-Tuning

With the models already trained on the aggregated dataset, the second and last training phase consisted of another fine-tuning, this time using the specific industrial dataset. The objective of this stage was to specialize the architectures for the scenario of adjacent boxes in crates by continuing the training process on this highly representative data.

The changes were applied specifically for each model. For the Faster R-CNN, the learning rate was maintained at $1 \times 10^{-4}$, a low value that allows for a more delicate adjustment of the weights. In addition, from tests, it was observed that the unfreezing of all layers brought greater learning gain. Due to the reduced size of the dataset (752 images), it was possible to extend the training for 100 epochs to ensure convergence.

For the YOLOv11x, the model was fine-tuned using the weights obtained from the previous training stage, with all layers unfrozen to allow full adaptation to the target domain. The optimization employed AdamW with an initial learning rate of $1 \times 10^{-2}$, linearly decayed to $1 \times 10^{-4}$ over 50 epochs, and included a 3-epoch warm-up phase. Training was conducted with a batch size of 8, input resolution of $640 \times 640$, momentum of 0.937, weight decay of $5 \times 10^{-5}$, and mixed precision enabled. In addition, data augmentation strategies such as horizontal flips, mosaic, random erasing, and RandAugment were applied to enhance robustness given the reduced dataset size.

The new configurations for this fine-tuning phase for both networks are summarized in Table 2.

## 4. Experiments

This section presents the experimental setup used to evaluate the performance of the proposed object detection models. The experiments aim to assess both the accuracy and efficiency of the models in detecting boxes in crates, using both generalist training and domain-specific fine-tuning. The evaluation considers standard detection metrics as well as inference speed

**Table 2.** Hyperparameters used in the second training phase for Faster R-CNN and YOLOv11x.

| Parameter | Faster R-CNN | YOLOv11x |
|---|---|---|
| Initial Weights | Stage 1 Weights | Stage 1 Weights |
| Optimizer | SGD | AdamW |
| Learning Rate | $1 \times 10^{-4}$ | $1 \times 10^{-4}$ |
| Batch Size | 4 | 8 |
| Epochs | 100 | 50 |
| Model Size | $\sim$ 158,04 MB | $\sim$ 114,4 MB |

to determine the suitability of each model for real-time automated inspection applications.

### 4.1 Evaluation Metrics

To quantitatively evaluate and compare the models on the test set, the following metrics were used:

- **Mean Average Precision (mAP@0.5):** Measures the precision of bounding box detection, considering a detection correct if the Intersection over Union (IoU) is greater than or equal to 0.5.
- **Mean Average Precision (mAP@0.75):** A stricter metric that considers a detection correct only if the IoU is greater than or equal to 0.75.

### 4.2 Experimental Environment

All experiments were conducted on two hardware platforms running Ubuntu, with the choice of platform based on the computational requirements of the model being evaluated:

- **Faster R-CNN:** Desktop with NVIDIA GeForce RTX 4090 GPU (24 GB VRAM), Intel Core i7-12700K CPU, and 128 GB RAM.
- **YOLOv11x:** Desktop with NVIDIA GeForce 4070 Ti GPU (12 GB VRAM), Intel i7-12700F CPU, and 64 GB RAM.

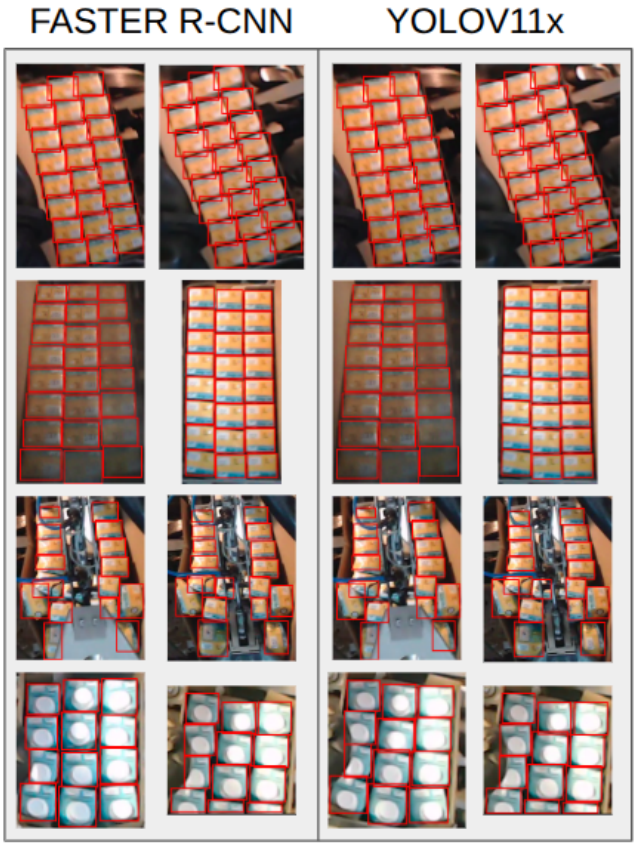


**Figure 3.** Qualitative comparison of Faster R-CNN and YOLOv11x detections on the test set. Both typical and challenging scenarios are shown to illustrate differences in model performance.

**Table 3.** Comparative performance results on the industrial test set.

| Metric | Faster R-CNN | YOLOv11x |
|---|---|---|
| mAP@.50 | 0.9890 | **0.9939** |
| mAP@.75 | 0.9662 | **0.9841** |

## 5. Results

To assess the effectiveness of the proposed two-phase training strategy, both models were systematically evaluated through quantitative and qualitative analyses. This section presents the experimental results obtained on the industrial dataset. The discussion that follows seeks not only to compare the numerical outcomes of Faster R-CNN and YOLOv11x, but also to interpret their practical implications for the detection of packages in crates within real industrial conditions.

### 5.1 Quantitative Analysis

After the conclusion of the second fine-tuning stage, both models were evaluated on the test set of the industrial dataset. Table 3 summarizes the main performance metrics for the Faster R-CNN and the YOLOv11x.

The results indicate that both models achieved exceptional performance in the task of detecting packages in crates, validating the effectiveness of the two-phase training strategy. With mAP@0.5 metrics superior to 98% for both architectures, the models' ability to locate the objects of interest with precision becomes evident.

The YOLOv11x model, however, established itself as superior in all accuracy metrics. It reached an mAP@0.75 of 0.9841, surpassing the 0.9662 of the Faster R-CNN, which denotes a greater robustness in more rigorous detection thresholds. The performance of the YOLOv11x at IoU=0.5 is almost perfect, with an mAP@0.5 of 99.40%. The evaluation on the industrial test set after the second fine-tuning stage demonstrates that both architectures achieved excellent detection performance, with mAP@0.5 greater than 0.98. Notably, YOLOv11x outperformed Faster R-CNN at stricter localization thresholds (mAP@0.75 of 0.9841 vs. 0.9662), indicating superior robustness under more demanding IoU criteria and a greater precision in bounding-box placement. These results corroborate the effectiveness of the two-phase fine-tuning strategy and suggest that the architectural and training differences of YOLOv11x provide an advantage for this specific industrial scenario. To complement these aggregate metrics, further analyses such as precision–recall curves and a detailed error analysis (false positives/false negatives) are recommended to fully characterise model behaviour and failure modes.

### 5.2 Qualitative Results

To complement the quantitative analysis and understand the performance differences between the models, a visual inspection of the detections on the test set was performed. Figure 3 illustrates good detections of both models in typical and challenging scenarios. To further exemplify model behavior within the production-line domain, eight representative images were selected, among them, the third row presents two distinct images per model showing a case in which the juices inside the crate have spilled, an especially challenging condition that nonetheless yields robust performance from both models.

In parallel, Figure 4 shows incorrect detections from both models, either predicting boxes where none exist or failing to detect some boxes.

Both models demonstrated an excellent ability to correctly identify and locate the packages in scenarios with full crates, which corroborates the high mAP@0.5 values obtained. However, the small discrepancies observed in the rigorous mAP metrics are explained by subtle differences in the precision of the bounding boxes or by failures in isolated cases of more complex detection.

Figure 5 shows a test scenario with high object density, where occlusion and the proximity between boxes required greater model precision. The right displays the detections of the YOLOv11x model, which managed to accurately delimit all packages, even those partially covered. In contrast, the left reveals a weak point of the Faster R-CNN, which, in this specific case, failed to detect one of the boxes and presented an imprecise bounding box for another, resulting in a lower score at the IoU threshold of 0.75.

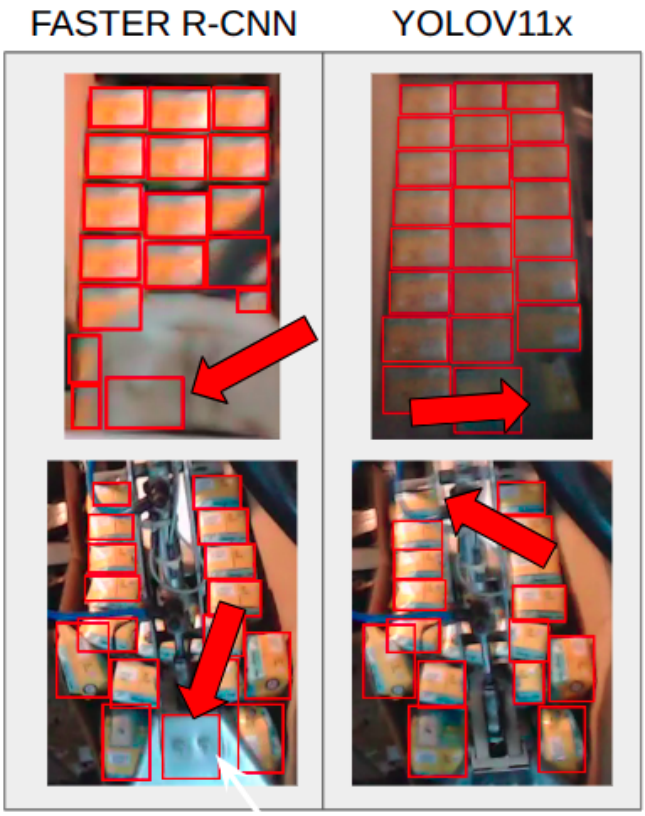


**Figure 4.** Examples where YOLO and FASTER made detection errors, either predicting boxes incorrectly or failing to detect them.

### 5.3 Conclusion

This work proposed a cost-aware, two-stage fine-tuning approach for automated detection of boxes in crates. In the first stage generalist model is trained on heterogeneous datasets to learn broad visual priors of the target objects. In the second stage the model is adapted to the production environment by fine-tuning on a compact, domain-specific dataset extracted from production-line videos and semi-automatically annotated with SAM. This staged strategy preserves generalization learned from diverse data while enabling precise adaptation to real operational conditions, substantially reducing manual annotation effort and improving robustness to variations in box placement, occlusion, and lighting.

Experiments with YOLOv11x and Faster R-CNN demonstrated the effectiveness of this strategy, with both models achieving mAP@0.5 above 98% and robust performance in scenarios with high object density. YOLOv11x outperformed Faster R-CNN in precision and inference speed, indicating its suitability for real-time industrial inspection.

Future work includes expanding the dataset to account for variations in lighting and reflections, as well as exploring lighter model architectures for deployment on cost-effective embedded systems. Overall, the proposed methodology provides an accurate, efficient, and practical solution for automated monitoring of secondary packaging in industrial environments.

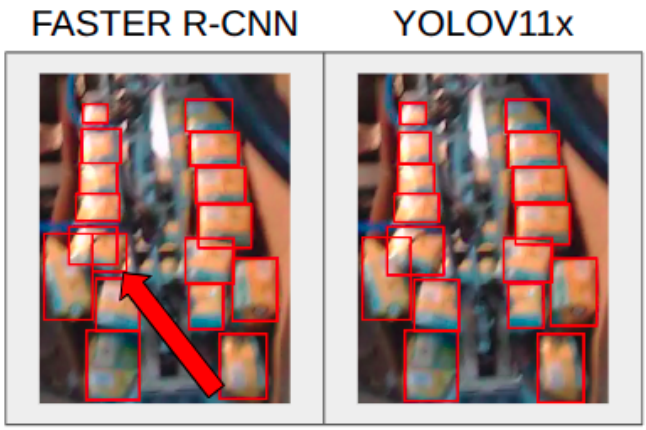


**Figure 5.** Comparison of Faster R-CNN and YOLOv11x detections on a challenging image. YOLO correctly predicts the bounding boxes, while Faster R-CNN duplicates one detection (indicated by the arrow).

## Acknowledgments

The authors would like to thank CAPES, CNPq, FAPEMIG and Tropical Industria de Alimentos S/A (Tial) for supporting this work. We are also grateful to Instituto de Inteligência Artificial e Computacional (Idata-UFV), FINEP (IA-AD-UFV #0284/22), and Cluster/UFV for providing GPU resources.

## Author Contributions

Gabriel was responsible for annotating frames for the specific dataset, implementing and conducting YOLO training and experiments, and contributing to the writing and figures of the manuscript. He also assisted with preprocessing, cleaning defective images, and aggregating data from the generalist Roboflow dataset. Ian managed dataset selection and preparation on Roboflow, implemented and conducted Faster R-CNN training and experiments, and contributed to the manuscript's text and figures. He also participated in selecting and annotating frames for the specific dataset. Michel M. Silva and Thiago L. Gomes served as consultants, providing critical feedback on the text and offering key ideas for the implementation of the main approach.

## References

[1] GHOBAKHLOO, M. Industry 4.0, digitization, and opportunities for sustainability. *Journal of Cleaner Production*, v. 252, p. 119869, 2020. ISSN 0959-6526. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/S0959652619347390⟩.

[2] ZRIBI, M.; PAGLIUCA, P.; PITOLLI, F. A computer vision-based quality assessment technique for the automatic control of consumables for analytical laboratories. *Expert Systems with Applications*, v. 256, p. 124892, 2024. ISSN 0957-4174. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/S0957417424017597⟩.

[3] KHAN, M. G. et al. Perfcam: Digital twinning for production lines using 3d gaussian splatting and vision models. *IEEE Access*, Institute of Electrical and Electronics Engineers (IEEE), v. 13, p. 85390–85406, 2025. ISSN 2169-3536. Disponível em: ⟨http://dx.doi.org/10.1109/ACCESS.2025.3567702⟩.

[4] A Review of Vision Based Defect Detection Using Image Processing Techniques for Beverage Manufacturing Industry. *Jurnal Teknologi (Sciences & Engineering)*, v. 81, n. 3, 2019. Disponível em: ⟨https://doi.org/10.11113/jt.v81.12505⟩.

[5] KIRILLOV, A. et al. *Segment Anything*. 2023. ArXiv preprint arXiv:2304.02643; Accessed: 2025-08-21. Disponível em: ⟨https://segment-anything.com⟩.

[6] JOCHER, G.; QIU, J. *Ultralytics YOLO11*. 2024. Disponível em: ⟨https://github.com/ultralytics/ultralytics⟩.

[7] REN, S. et al. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. 2016. Disponível em: ⟨https://arxiv.org/abs/1506.01497⟩.

[8] DWYER, B. et al. *Roboflow (Version 1.0) [Software]*. 2024. Computer vision. Disponível em: ⟨https://roboflow.com⟩.

[9] ANGELOLMG. Open Source Dataset, *milk-detector-v2 Dataset*. Roboflow, 2022. Accessed: 2025-07-03. Disponível em: ⟨https://universe.roboflow.com/angelolmg/milk-detector-v2⟩.

[10] 1, S. test. Open Source Dataset, *sort the trash V4 Dataset*. Roboflow, 2023. Accessed: 2025-07-04. Disponível em: ⟨https://universe.roboflow.com/sampahsem5-test-1/sort-the-trash-v4⟩.

[11] JAHANGIR, S. Open Source Dataset, *Slice_Juice Dataset*. Roboflow, 2024. Accessed: 2025-07-04. Disponível em: ⟨https://universe.roboflow.com/sarmad-jahangir-ph67v/slice_juice⟩.

[12] LONKYKONG. Open Source Dataset, *juice-detection-4-items Dataset*. Roboflow, 2023. Accessed: 2025-07-04. Disponível em: ⟨https://universe.roboflow.com/lonkykong/juice-detection-4-items⟩.

[13] LEARNING, M. Open Source Dataset, *Grocery Items Dataset*. Roboflow, 2024. Accessed: 2025-07-04. Disponível em: ⟨https://universe.roboflow.com/machine-learning-chipg/grocery-items-re4qf⟩.

[14] University. Open Source Dataset, *RetinaNet Dataset*. Roboflow, 2024. Accessed: 2025-07-04. Disponível em: ⟨https://universe.roboflow.com/university-d2l2p/retinanet-besb5⟩.

[15] YANG, J. et al. Scd: A stacked carton dataset for detection and segmentation. *Sensors*, v. 22, n. 10, 2022. ISSN 1424-8220. Disponível em: ⟨https://www.mdpi.com/1424-8220/22/10/3617⟩.