YOLO-Guided Pretraining for Apple Counting under Limited Annotations *

Giuliano Ramírez[1†], Orietta Nicolis[1,3], Hans Lobel[2], Billy Peralta[1,4†]

[1] Facultad de Ingeniería, Universidad Andres Bello, Santiago, Chile
[2] Escuela de Ingeniería, Pontificia Universidad Católica de Chile, Santiago, Chile
[3] Department of Economics, University of Messina, Messina, Italy
[4] Transportation and Logistics Center, Universidad Andres Bello, Santiago, Chile
g.ramrezprez@uandresbello.edu, orietta.nicolis@unab.cl, hlobel@ing.puc.cl, billy.peralta@unab.cl

Abstract

Precision agriculture requires scalable and data-efficient fruit counting, as manual tallies are slow and vision-based models still rely heavily on fully labeled datasets. We introduce a pseudo/semi-supervised pipeline where a YOLO detector generates pseudo-labels to pretrain a U-Net segmenter. Specifically, we treat YOLO as a teacher to pretrain a specialized counting model, transferring detection cues into a segmentation-based counter. On the MinneApple benchmark, this approach reduces average error from RMSE 15.35 and MAE 11.50 to 13.03 and 10.10, while raising R^2 from 0.66 to 0.77, with stable behavior around detector thresholds of 0.065–0.075. When labeled data are reduced (e.g., from 67/13/10 to 17/3/10 splits), the pretrained model consistently outperforms the baseline, increasing R^2 by more than 0.12 in the most label-scarce regime. Scaling pseudo-labeled pretraining while reducing supervised data by 25–75% further improves performance, with R^2 gains up to 0.71 compared to 0.56. These findings demonstrate that detector-driven pretraining is an effective and label-efficient strategy for fruit counting under data scarcity, and point toward extensions across crop types, sensing modalities, and deployment-oriented advances such as adaptive thresholding, active learning, and domain adaptation.

Introduction

In many agricultural supply chains, accurate fruit counting underpins harvest planning, logistics, storage, and yield estimation; however, counts are still frequently gathered by hand, which is slow and error-prone. Recent progress in computer vision (CV) for agriculture such as spanning disease detection, growth monitoring, and even automated harvesting, suggests that reliable, scalable fruit counting should be feasible, yet in practice most CV pipelines still depend on large, fully labeled datasets that are costly to produce and hard to maintain across orchards, seasons, lighting, and sensor variations (Kamilaris and Prenafeta Boldú 2018). This creates a structural bottleneck: data annotation rather than modeling capacity limits deployment. Moreover, existing systems such as DeepFruits and MinneApple highlight two persistent hurdles: illumination variability and domain specificity, where models perform best only near the conditions seen at training time, restricting real-world generalization.

A recent comprehensive review by (Lv et al. 2025) shows that, despite advances in deep learning, apple growth monitoring and fruit yield estimation systems still struggle with occlusion, changing illumination, and orchard variability. These limitations highlight the need for label-efficient approaches that lower annotation costs while maintaining robustness in complex orchard environments. Building on this context, our work addresses data scarcity in fruit counting through methods that minimize dependence on human labels and leverage modern detectors and segmenters to transfer knowledge effectively.

The core technical challenge we address is the dependence of prevailing fruit-counting methods on dense, human-verified labels,an assumption that often fails in orchard imagery characterized by partial occlusions, inconsistent lighting, and cluttered backgrounds. Annotation in these settings is not only expensive but also noisy and inconsistent, further constraining downstream model performance and scalability. While the literature frequently assumes the availability of fully labeled datasets, production environments rarely match this ideal: most growers cannot afford the level of annotation required to capture cultivar differences, growth stages, and sensor setups. Solutions must therefore learn reliably from a small number of labeled images and a much larger pool of unlabeled ones. Consequently, a natural question is: *can a high-capacity detector serve as a teacher to pretrain a specialized counter when labels are scarce?*. We explicitly pursue a label-efficient strategy that treats detector outputs as a surrogate supervision signal, aiming to reduce human effort without sacrificing accuracy. By framing detection-driven pretraining as a bridge between low-cost pseudo-labels and a segmentation-based counter, we mitigate both the annotation bottleneck and the brittleness to environmental factors documented in prior sys-

*Accepted at the First International Workshop on AI in Agriculture (Agri AI), co-located with AAAI 2026.

†These authors contributed equally.

tems.

Concretely, we propose a pseudo/semi-supervised pipeline in which a real-time object detector (YOLO) is used as a surrogate task to generate masks on unlabeled orchard images, transferring this signal to a U-Net-based counting model via pretraining. To select reliable pseudo-labels, we sweep detector confidence thresholds over a fine grid (0.005–0.1) and evaluate across multiple random seeds to stabilize selection; we then train with stratified data distributions that progressively reduce the fraction of supervised data and increase the fraction of pseudo-labeled data. Our experiments use the MinneApple dataset of apple-tree images (initial resolution 720×1280), with 671 labeled and 332 unlabeled samples; for efficiency we downsample to 360×640 while preserving counting fidelity. Models are assessed under Random Holdout using RMSE, MAE, and R^2, and we analyze performance across seeds, thresholds, and labeled/unlabeled splits to quantify label-efficiency and robustness. This design reflects a practical deployment scenario where a small curated labeled set must bootstrap a much larger, heterogeneous pool of orchard imagery, and where thresholding and pretraining act as dials to trade off label cost and accuracy.

The contributions of this paper are threefold. First, it formalizes a detection-to-segmentation pretraining pipeline for fruit counting under data scarcity, detailing an algorithmic procedure (Algorithm 1) that integrates threshold selection, seed averaging, and curriculum-style data distributions, and reporting all implementation choices for reproducibility. Second, it presents a rigorous experimental study on MinneApple that varies labeled/unlabeled proportions, thresholds, and seeds, and evaluates with RMSE/MAE/R^2 under K-Random Holdout. Third, it provides empirical evidence that YOLO-driven pretraining consistently improves UNet counting performance relative to a supervised-only baseline across multiple settings, underscoring intelligent pretraining and pseudo-labeling as practical levers for deployment. The remainder of the paper reviews related work, describes the proposed method and training schedule, and then reports quantitative and qualitative results before closing with limitations and future directions.

Related Work

Research on fruit counting and object density estimation has attracted considerable attention in recent years, motivated both by agricultural applications and by advances in crowd and object counting in computer vision. Broadly, existing works fall into two categories: (a) detection-based pipelines that identify individual objects before aggregation, and (b) segmentation- or density-based methods that estimate counts from continuous representations. Within agriculture, detection-driven approaches have dominated, while density-based frameworks have gained traction in crowd counting and are increasingly being adapted for plant and fruit analysis.

Bargoti and Underwood (2016) pioneered the use of deep convolutional networks for fruit detection in orchards. Their system demonstrated that robust fruit detection could be achieved under natural lighting conditions, but the model was heavily reliant on large amounts of labeled data. This highlighted the annotation bottleneck and motivated the search for methods capable of reducing supervision while preserving accuracy in orchard environments.

Sa et al. (2016) introduced DeepFruits, a fruit detection system based on deep neural networks that integrated multiple sensor modalities. Their work showed significant improvements in recall and precision for fruit detection, but their approach was limited by domain specificity—models needed to be retrained or fine-tuned when deployed in different orchard settings. This underscored the challenges of scalability and domain adaptation in agricultural computer vision.

Rahnemoonfar and Sheppard (2017) proposed Deep Count, a simulated learning framework for fruit counting that avoided explicit bounding box annotations. By training on synthetically generated images, they reduced labeling requirements, but the reliance on simulation constrained applicability to real-world conditions where variation is much higher than in synthetic domains. Nevertheless, the study demonstrated the feasibility of learning counting tasks with minimal real annotations.

Häni, Roy, and Isler (2020) released the MinneApple dataset, a benchmark collection of labeled orchard images for fruit detection and segmentation. This dataset has become central to subsequent research, offering standardized benchmarks and enabling reproducible comparisons across methods. However, despite its utility, annotation costs remained significant and dataset scale was still modest compared to the requirements of modern deep learning models.

Koirala et al. (2019) developed MangoYOLO, a YOLO-based detector adapted for mango yield estimation. They benchmarked real-time detection on orchard imagery and proposed a pipeline for fruit load estimation. Their approach underscored the efficiency of detection-driven systems but also reaffirmed their dependence on carefully curated training data and domain-specific tuning, again raising the issue of scalability across crops and conditions.

More broadly, computer vision research on object counting has informed agricultural applications. Loy, Gong, and Xiang (2013) advanced semi-supervised crowd counting, demonstrating that transfer learning can mitigate label scarcity, while Boominathan, Kruthiventi, and Babu (2016) introduced CrowdNet, a density-map CNN for dense crowd counting, showing how continuous maps can outperform detection in cluttered scenes. Similarly, Xiong et al. (2019) refined divide-and-conquer strategies for closed-set object counting, illustrating scalable principles that could inform agricultural counting tasks.

Other surveys and reviews provide broader context. Lu and Young (2020) catalogued public datasets for precision agriculture, highlighting the gap in large-scale annotated data, while Zhang et al. (2020) reviewed deep learning approaches for dense agricultural scenes, identifying annotation cost and generalization as central open challenges. Heinrich, Roth, and Zschech (2019) offered a taxonomy of object counting strategies across domains, providing a conceptual framework to situate fruit counting within general object density estimation.

Zhang et al. (2021) address the species/domain gap in fruit detection by proposing a lightweight domain-adaptation scheme (EasyDAM) that leverages minimal target supervision with pseudo-labels and distribution alignment. The method improves mAP when transferring detectors across orchards and fruit species, especially under illumination/background shifts, while avoiding full re-annotation.

Gao et al. (2022) present a detection-and-counting pipeline for apples that couples a CNN-based detector with a trunk-tracking module enforcing row-wise geometric consistency. Performance is sensitive to planting geometry and camera viewpoint, yet the results illustrate how structural priors can stabilize counting beyond pure detection or density-map baselines.

In more recent work, Johanson et al. (2024) present S3AD, a semi-supervised framework for *small apple detection* that leverages both labeled and unlabeled orchard images, improving detection under data scarcity and varying illumination; their results underscore the value of semi-supervised signals when annotation budgets are tight.

Hu et al. (2023) provide a detection-and-counting pipeline for apples in orchards and a careful comparison of detection-, regression-, and density-based strategies, finding detection-driven methods generally superior for real-world clutter—an observation aligned with detection-to-segmentation pretraining schemes.

Zhang et al. (2024) propose *DomAda-FruitDet*, a domain-adaptive, anchor-free fruit detector with auto-labeling, explicitly tackling domain shift across orchards and showing how auto/pseudo-labels can bootstrap robust detectors, which dovetails with label-efficient pretraining.

Li et al. (2025) introduce *MetaFruit*, a more diverse multi-fruit dataset and protocol that engage recent foundation-model practices, highlighting the benefits of larger, heterogeneous corpora for transfer and pretraining in agricultural vision.

Overall, prior work has shown that deep learning is effective for fruit detection and counting, yet most methods depend heavily on fully labeled data, struggle with generalization across orchard conditions, or rely on simulation-based training that diverges from real-world variability. In contrast, our work introduces a pseudo/semi-supervised pipeline where a detector generates pseudo-labels that pretrain a segmentation-based counter, systematically reducing annotation cost while retaining accuracy.

Proposed Method

We propose a detector-to-segmenter pipeline that leverages a real-time object detector to generate pseudo-labels on unlabeled orchard images and transfers this signal to a segmentation-based counter via pretraining. The approach is tailored to orchard conditions (occlusions, variable illumination, and clutter) and explicitly targets label efficiency: the detector supplies objectness priors at scale, while the segmenter consolidates fine structures beneficial for counting. Empirically, this combination consistently improves the segmentation model relative to a supervised-only UNet baseline under data scarcity.

Problem Setup and Notation

Let $\mathcal{D}_L = \{(x_i, y_i)\}_{i=1}^{N_L}$ be a labeled set of orchard images with pixel-level masks y_i (for counting/segmentation) and $\mathcal{D}_U = \{x_j\}_{j=1}^{N_U}$ an unlabeled pool. A detector f_{det} (YOLO-like) produces per-instance predictions on $x \in \mathcal{D}_U$, which we rasterize to a binary mask $\tilde{y}$ indicating fruit pixels. A segmentation model f_{seg} (UNet-like) is then (i) *pretrained* on pseudo-masks from $\mathcal{D}_U$ and (ii) *fine-tuned* on $\mathcal{D}_L$. To reduce compute while preserving counting fidelity, we downsample all images from the original resolution to 360×640 and adopt a Random Holdout protocol with disjoint Train/-Valid/Test splits. We report RMSE, MAE, and R^2 on the held-out test set.

Pseudo-Label Generation via Threshold Sweep and Seed Averaging

The quality of pseudo-labels is controlled by a detector confidence threshold τ. We sweep τ over a fine grid, $\tau \in \{0.005, 0.010, \ldots, 0.100\}$, and evaluate each candidate on a validation portion of $\mathcal{D}_L$ by comparing detector-derived masks $\tilde{y}_\tau$ with ground-truth masks y (a mask-agreement score such as IoU/Dice or an error proxy aligned with counting). To stabilize selection, we repeat the process across multiple random seeds $s \in \mathcal{S}$, shuffling data before inference and averaging validation scores:

$$\tau^\star \;=\; \arg\max_{\tau} \; \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \mathrm{Score}\big(\tilde{y}_{\tau,s},\, y\big).$$

The resulting $\tau^\star$ balances precision/recall in pseudo-labels, reducing noise in the subsequent pretraining stage. Pseudo-masks from $\mathcal{D}_U$ at $\tau^\star$ are cached to avoid recomputation and to ensure deterministic pretraining runs.

Curriculum over Labeled/Unlabeled Splits

We adopt a curriculum that progressively increases reliance on pseudo-labeled data while reducing human-labeled supervision. Let π_{train} and π_{valid} denote the percentages of $\mathcal{D}_L$ allocated to Train and Valid, respectively, with Test fixed. We consider the sequence of splits: 75/15, 67/13, 50/10, 33/7, 17/3 (train/valid considering a total possible of 90%; test fixed at 10%). For each distribution, we (1) shuffle labeled and unlabeled sets with seed s, (2) produce pseudo-masks on $\mathcal{D}_U$ at $\tau^\star$, (3) pretrain f_{seg} on $\mathcal{D}_U$ with pseudo-masks, and (4) fine-tune on $\mathcal{D}_L$ (Train), selecting the model by Valid. This schedule quantifies label efficiency directly and tests robustness across seeds. Figure 1 shows a summary of proposed workflow.

Training Schedule and Algorithm

The training routine follows a two-stage schedule repeated for each seed s and distribution:

1. **Pretraining on $\mathcal{D}_U$:** Initialize f_{seg}; train with pseudo-masks $\tilde{y}_{\tau^\star}$ to learn orchard-specific features from unlabeled data. Basic augmentations (resize, flips, mild color jitter) are applied consistently across stages to mitigate overfitting.

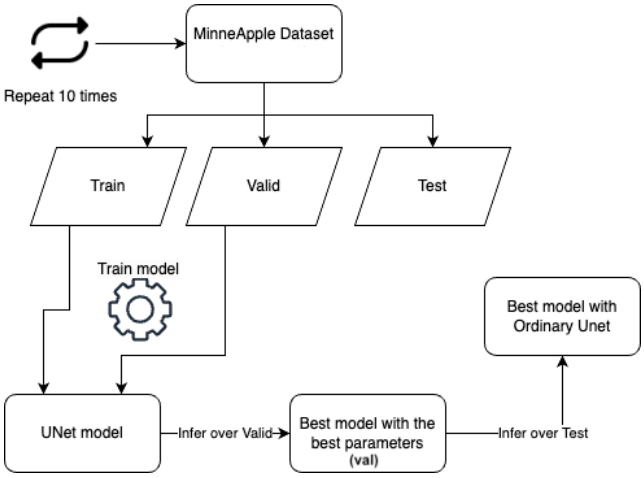

Figure 1: Experimental workflow.

Figure 2: Sample images from MinneApple dataset.

2. **Supervised Fine-tuning on $\mathcal{D}_L$:** Fine-tune f_{seg} on pixel-level masks y (Train) and select checkpoints by Valid. Early stopping and a small learning-rate decay help stabilize convergence after pretraining.

The overall procedure is summarized below, mirroring the implementation used in our experiments:

Experimental Context

Data. We use the MinneApple dataset (apples on trees) with 671 labeled and 332 unlabeled images. All images are downscaled to 360×640 to speed up training while retaining adequate spatial detail for counting. Figure 2 show several sample images of this dataset.

Evaluation. We employ Random Holdout to form Train/-Valid/Test for each curriculum point and seed. Metrics are RMSE, MAE, and R^2 reported on the test split; per-seed statistics are aggregated to reduce variance. We share a demo code of this work at https://drive.google.com/drive/folders/18HpD7JlJ48REbArWUqDU192A2gZVEL-d

Why detector→U-Net model? Detectors offer strong instance-level priors (objectness, scale) that are particularly helpful to bootstrap learning on unlabeled data, whereas the U-Net model consolidates fine boundaries and pixel continuity that detectors may miss in cluttered foliage. This complementarity drives the consistent gains observed after pretraining relative to a segmentation-only baseline under reduced supervision.

Algorithm 1: Detector-Driven Pretraining for Fruit Counting

Inputs: $\mathcal{D}_L, \mathcal{D}_U$, seeds $\mathcal{S}$, threshold grid $\mathcal{T} = \{0.005, \ldots, 0.100\}$, distributions $\mathcal{P} = \{(75,15), (67,13), (50,10), (33,7), (17,3)\}$
function SELECTTHRESHOLD($\mathcal{D}_L, \mathcal{T}, \mathcal{S}$)
  for $s \in \mathcal{S}$ **do**
    shuffle $\mathcal{D}_L$ with s;
    for $\tau \in \mathcal{T}$ **do**
      infer detector on $\mathcal{D}_L$; build $\tilde{y}_{\tau,s}$;
      score = mask-agreement$(\tilde{y}_{\tau,s}, y)$; cache (τ, s, score);
    end for
  end for
  return $\tau^\star = \arg\max_\tau \frac{1}{|\mathcal{S}|} \sum_s \text{score}(\tau, s)$
end function
$\tau^\star \leftarrow$ SELECTTHRESHOLD($\mathcal{D}_L, \mathcal{T}, \mathcal{S}$)
for $s \in \mathcal{S}$ **do**
  shuffle $(\mathcal{D}_L, \mathcal{D}_U)$ with s;
  for $(p_{\text{train}}, p_{\text{valid}}) \in \mathcal{P}$ **do**
    split $\mathcal{D}_L \rightarrow$Train($p_{\text{train}}$), Valid($p_{\text{valid}}$), Test(10%);
    build pseudo-masks on $\mathcal{D}_U$ with $\tau^\star$;
    pretrain f_{seg} on $(\mathcal{D}_U, \tilde{y}_{\tau^\star})$;
    fine-tune on Train; select by Valid; evaluate on Test (RMSE, MAE, R^2).
  end for
end for

Table 1: Seed-averaged performance for the threshold sweep. Means are taken across 10 seeds.

Setting	RMSE ↓	MAE ↓	R^2 ↑
Original (mean across seeds)	15.35	11.50	0.66
PreTrain @ 0.065	15.65	11.66	0.66
PreTrain @ 0.070	**13.03**	**10.10**	**0.77**
PreTrain @ 0.075	14.30	11.00	0.71
PreTrain (avg over thresholds)	14.32	10.92	0.71

Results

Setup. We report counting accuracy using RMSE, MAE (lower is better), and R^2 (higher is better). Results are aggregated across repeated random seeds. We first sweep three detector confidence thresholds to generate pseudo-masks for pretraining a U-Net-based counting model, then study label-efficiency under progressively smaller supervised splits, and finally assess how scaling pseudo-labeled pretraining data while reducing supervised data impacts accuracy.

Global analysis. Averaged across seeds, the best pseudo-label threshold is 0.070, reducing RMSE from 15.35 to 13.03 (-15.1%) and MAE from 11.50 to 10.10 (-12.2%), while increasing R^2 from 0.66 to 0.77 ($+16.9\%$). Averaging the three thresholds still yields a consistent gain over

Table 2: Label-efficiency study: seed-averaged absolute performance by train/valid split (test remains 10%). No seed or threshold is shown; rows are means across all available seeds.

Original			
Split (train/valid/test%)	RMSE	MAE	R^2
75/15/10	15.35	11.50	0.66
67/13/10	14.64	11.38	0.67
50/10/10	15.47	12.19	0.73
33/7/10	16.03	12.59	0.68
17/3/10	18.22	14.55	0.52
PreTrain			
Split (train/valid/test%)	RMSE	MAE	R^2
75/15/10	**12.53**	**9.36**	**0.79**
67/13/10	**13.51**	**10.54**	**0.74**
50/10/10	**14.39**	**11.04**	**0.77**
33/7/10	**13.36**	**10.25**	**0.77**
17/3/10	**15.54**	**12.26**	**0.64**

Table 3: Scaling pseudo-labeled pretraining while reducing supervised data. Means across seeds; no seed/threshold shown.

Original			
Pretrain ↑/Supervised ↓	RMSE	MAE	R^2
$+25\%/-25\%$	14.87	10.79	0.69
$+50\%/-50\%$	17.60	13.73	0.59
$+75\%/-75\%$	16.55	12.56	0.56
PreTrain			
Pretrain ↑/Supervised ↓	RMSE	MAE	R^2
$+25\%/-25\%$	**13.33**	**10.15**	**0.76**
$+50\%/-50\%$	**14.80**	**10.93**	**0.69**
$+75\%/-75\%$	**13.38**	**10.14**	**0.71**

the non-pretrained model (-6.7% RMSE, -5.1% MAE, $+8.0\%$ R^2).

Sensitivity to the amount of labeled data Pretraining improves the mean across all splits. The gains become more pronounced as labeled data shrinks: at 33/7/10 we observe -16.6% RMSE, -18.6% MAE and $+14.7\%$ R^2; at 17/3/10 we obtain -14.7%, -15.7%, and $+24.3\%$ respectively (relative to the corresponding original means).

Sensitivity to pseudo-labeled data. Increasing the pseudo-labeled pretraining pool while decreasing supervised data consistently improves absolute accuracy: $+25/-25$ yields -10.4% RMSE, -5.9% MAE, and $+9.4\%$ R^2; $+75/-75$ strengthens this to -19.2%, -19.3%, and $+28.2\%$.

Main Insights. In general, the detector-driven pretraining with a threshold around 0.07 delivers the best mean accuracy and generalizes robustly as labeled supervision diminishes. The improvements are largest in the most label-scarce regimes, indicating a strong practical path toward scalable fruit counting with limited annotation budgets.

Discussion

The aggregated experiments yield three high-level observations. First, detector-driven pretraining consistently improves the UNet baseline across thresholds, with the best performance obtained at a detector confidence of 0.070 (RMSE 13.03, MAE 10.10, R^2 0.77) compared to the original averages (RMSE 15.35, MAE 11.50, R^2 0.66). This confirms that objectness priors distilled from a detector provide a strong inductive bias for dense counting. Second, the relative advantage of pretraining becomes more pronounced as the amount of labeled data decreases. For instance, when supervision is reduced to $33/7/10$, the pretrained model achieves (RMSE 13.36, MAE 10.25, R^2 0.77) versus the baseline (RMSE 16.03, MAE 12.59, R^2 0.68), and at $17/3/10$ the gap widens further (R^2 rising from 0.52 to 0.64). Third, scaling pseudo-labeled pretraining while shrinking the supervised set yields monotonic benefits: under $+75/-75$ the pretrained model achieves (RMSE 13.38, MAE 10.14, R^2 0.71) versus the baseline (RMSE 16.55, MAE 12.56, R^2 0.56), suggesting a clear deployment path when annotation budgets are limited.

We also observe outliers. Although the average improvements are consistent, certain seeds or suboptimal thresholds can underperform relative to the baseline, as seen in individual rows at the 75/15/10 or 50/10/10 splits. This highlights the sensitivity of pseudo-label quality to the detector threshold and to seed-dependent permutations. Such variance underscores the need for principled threshold selection and potentially lightweight denoising of pseudo-masks (e.g., small connected-component filtering or morphological operations) to reduce label noise.

Limitation. We note that a few seeds underperform the baseline, suggesting sensitivity to pseudo-label quality; future work will study these failure cases and adopt uncertainty-aware filtering to improve robustness.

Practical implications. Given the strong signal from both label-efficiency and pretraining-scale studies, orchard deployments should prioritize collecting large pools of unlabeled data and leverage detector-guided pseudo-labels for pretraining dense counters, while curating a modest, high-quality labeled set for fine-tuning. The combination of a threshold sweep and seed averaging provides a lightweight, AutoML-like recipe to stabilize results across random initializations.

Conclusions

Detector-guided pretraining proves to be a practical and robust strategy for fruit counting under data scarcity. The method consistently improves absolute accuracy: at the optimal threshold, RMSE drops from 15.35 to 13.03, MAE from 11.50 to 10.10, and R^2 rises from 0.66 to 0.77; under reduced supervision (e.g., $17/3/10$) the pretrained model lifts R^2 by more than 0.12; and when pseudo-labeled pretraining is scaled ($+75/-75$), accuracy improves by nearly three RMSE points and R^2 rises from 0.56 to 0.71. These findings confirm that leveraging detector objectness to seed a dense counter can materially reduce the need for exhaustive manual annotation without sacrificing accuracy.

Future work should (i) adopt adaptive thresholding or teacher confidence calibration to limit pseudo-label noise, (ii) integrate simple mask denoising and uncertainty-weighted losses, and (iii) extend beyond apples to multi-crop settings and additional sensing modalities. Testing domain adaptation routines alongside active learning for sample-efficient labeling will further advance deployment in real agricultural environments.

Acknowledgments

O.Nicolis and B.Peralta acknowledge support from ANID – Fondecyt grants 1241881 and 1241882. H.Lobel and B.Peralta also acknowledge support of the National Center for Artificial Intelligence CENIA FB210017, Basal ANID.

References

Bargoti, S.; and Underwood, J. 2016. Deep fruit detection in orchards. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 3626–3633.

Boominathan, L.; Kruthiventi, S.; and Babu, R. V. 2016. CrowdNet: A Deep Convolutional Network for Dense Crowd Counting. In *Proceedings of the 24th ACM International Conference on Multimedia*, 640–644.

Gao, F.; Fang, W.; Sun, X.; Wu, Z.; Zhao, G.; Li, G.; Li, R.; Fu, L.; and Zhang, Q. 2022. A novel apple fruit detection and counting methodology based on deep learning and trunk tracking in modern orchard. *Computers and Electronics in Agriculture*, 197: 107000.

Heinrich, K.; Roth, A.; and Zschech, P. 2019. Everything counts: a Taxonomy of Deep Learning Approaches for Object Counting. In *ECIS*.

Hu, J.; Fan, C.; Wang, Z.; Ruan, J.; and Wu, S. 2023. Fruit detection and counting in apple orchards based on improved Yolov7 and multi-Object tracking methods. *Sensors*, 23(13): 5903.

Häni, N.; Roy, P.; and Isler, V. 2020. MinneApple: A benchmark dataset for apple detection and segmentation. *IEEE Robotics and Automation Letters*, 5(2): 852–858.

Johanson, R.; Wilms, C.; Johannsen, O.; and Frintrop, S. 2024. S^3AD: Semi-Supervised Small Apple Detection in Orchard Environments. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 7076–7085.

Kamilaris, A.; and Prenafeta Boldú, F. 2018. Deep Learning in Agriculture: A Survey. *Computers and Electronics in Agriculture*, 147.

Koirala, A.; Walsh, K. B.; Wang, Z.; and McCarthy, C. 2019. Deep learning for real-time fruit detection and orchard fruit load estimation: Benchmarking of 'MangoYOLO'. *Precision Agriculture*, 20(6): 1107–1135.

Li, J.; Lammers, K.; Yin, X.; Yin, X.; He, L.; Sheng, J.; Lu, R.; and Li, Z. 2025. MetaFruit meets foundation models: Leveraging a comprehensive multi-fruit dataset for advancing agricultural foundation models. *Computers and Electronics in Agriculture*, 231: 109908.

Loy, C. C.; Gong, S.; and Xiang, T. 2013. From semi-supervised to transfer counting of crowds. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2256–2263.

Lu, Y.; and Young, S. 2020. A survey of public datasets for computer vision tasks in precision agriculture. *Computers and Electronics in Agriculture*, 178: 105760.

Lv, M.; Xu, Y.-X.; Miao, Y.-H.; and Su, W.-H. 2025. A Comprehensive Review of Deep Learning in Computer Vision for Monitoring Apple Tree Growth and Fruit Production. *Sensors*, 25(8): 2433.

Rahnemoonfar, M.; and Sheppard, C. 2017. Deep count: Fruit counting based on deep simulated learning. *Sensors*, 17(4): 905.

Sa, I.; Ge, Z.; Dayoub, F.; Upcroft, B.; Perez, T.; and McCool, C. 2016. DeepFruits: A fruit detection system using deep neural networks. *Sensors*, 16(8): 1222.

Xiong, H.; Lu, H.; Liu, C.; Liu, L.; Cao, Z.; and Shen, C. 2019. From open set to closed set: Counting objects by spatial divide-and-conquer. *IEEE Transactions on Image Processing*, 28(11): 5446–5460.

Zhang, Q.; Liu, Y.; Gong, C.; Chen, Y.; and Yu, H. 2020. Applications of deep learning for dense scenes analysis in agriculture: A review. *Sensors*, 20(5): 1520.

Zhang, W.; Chen, K.; Wang, J.; Shi, Y.; and Guo, W. 2021. Easy domain adaptation method for filling the species gap in deep learning-based fruit detection. *Horticulture Research*, 8.

Zhang, W.; Zheng, C.; Wang, C.; and Guo, W. 2024. DomAda-FruitDet: Domain-Adaptive Anchor-Free Fruit Detection Model for Auto Labeling. *Plant Phenomics*, 6: 0135.