

AGRICOUNTDINO: PARAMETER-EFFICIENT EXEMPLAR-GUIDED COUNTING AND LOCALIZATION IN AGRICULTURE

Shengjie Guo^{1,3}, Xin Li², Borjana Arsova³, Hanno Scharr⁴, Silvio Salvi^{1*}

¹Department of Agricultural and Food Sciences, University of Bologna, Bologna, Italy

²Department of Architecture, Built Environment and Construction Engineering, Politecnico di Milano, Milan, Italy

³Institute of Bio- and Geosciences, Plant Sciences, Forschungszentrum Jülich, Jülich, Germany

⁴Institute for Advanced Simulation, Data Analytics and Machine Learning, Forschungszentrum Jülich, Jülich, Germany

ABSTRACT

Accurate counting and localization of plants and their organs support phenotyping and yield estimation, yet target appearance, scale, and density vary widely across species and imaging conditions. Exemplar boxes specify the target without category-specific retraining, and point predictions identify the individual instances contributing to the count. We introduce AgriCountDINO, a parameter-efficient exemplar-guided framework for joint counting and localization. It conditions frozen multiscale DINOv3 features on exemplar appearance and size, then progressively decodes them into target points. Missed-object recovery extends supervision to targets overlooked by initial matching, and exemplar-adaptive point NMS filters duplicate predictions according to exemplar scale. With 8.4M trainable parameters, approximately one-tenth of TasselNetV4's, AgriCountDINO achieves a three-shot MAE of 11.92 on the TPC-268 benchmark, reducing counting error by 9.7% while providing individual target locations. Trained only on TPC-268, it achieves a zero-shot MAE of 14.25 on unseen generic object categories in FSC-147, improving upon the best compared zero-shot method by 6.0% without target-domain training or fine-tuning.

Index Terms— Plant Agnostic Counting, Exemplar-guided Counting, Point Localization, Computer Vision

1. INTRODUCTION

Accurate counting and localization of plants and their organs are fundamental to agricultural image analysis. Count estimates support yield estimation and phenotyping, while instance locations enable spatial analysis and inspection of individual predictions. Agricultural targets vary substantially across species, organs, growth stages, scales, and imaging conditions, making it difficult to extend category-specific models to diverse targets [1]. This diversity motivates exemplar-guided approaches, in which a few visual examples specify the objects of interest without requiring a separate model for each target category.

Recent counting methods can return individual target locations through point predictions, detections, or segmentation masks [2, 3, 4, 5]. In agriculture, TasselNet introduced local count regression for maize tassels [6], while TasselNetV4 combined local counting with exemplar guidance to address variation across scenes, scales, and species [1]. TPC-268 provides exemplar boxes and point annotations for hundreds of fine-grained plant categories [7], allowing counting and localization to be evaluated together across diverse agricultural targets.

TasselNetV4's local-count predictions, however, do not explicitly identify individual targets. Exemplar-guided counting depends on relating the supplied examples to corresponding regions in the image [8]. To also localize targets, these correspondences must retain the spatial detail needed to distinguish nearby instances as plant appearance and size vary. Self-supervised visual encoders provide transferable representations [9, 10]; DINOv3, in particular, offers dense features that provide a basis for learning such correspondences [11]. This motivates **AgriCountDINO**¹, an exemplar-guided framework for joint plant counting and localization.

AgriCountDINO uses a frozen DINOv3-pretrained encoder [11, 12] to extract multiscale image features. Exemplar appearance and size condition these features, which are fused from coarse to fine and decoded into instance points. Training updates only the exemplar-conditioning, fusion, and prediction modules. Missed-object recovery supervises annotated targets overlooked by initial peak matching. At inference, the exemplar scale determines which nearby point predictions are suppressed as duplicates. The retained points specify individual target locations, and their number gives the count. Counting and point-localization results on diverse plant categories, together with zero-shot transfer to unseen generic objects, suggest that the model captures transferable visual and semantic correspondences between exemplars and targets.

Our contributions are summarized as follows:

- We introduce AgriCountDINO, an exemplar-guided model that counts and localizes diverse plants and plant organs with only 8.4M trainable parameters.
- We develop multiscale point prediction guided by exemplar appearance and size, with missed-object recovery during training and exemplar-adaptive duplicate suppression at inference.
- AgriCountDINO achieves a three-shot validation MAE of 11.92 on TPC-268, 9.7% below TasselNetV4, and a zero-shot MAE of 14.25 on FSC-147, 6.0% below the strongest compared zero-shot method.

2. METHOD

Given an image I and a set of exemplar boxes $\mathcal{E}$, AgriCountDINO predicts target locations $\hat{\mathcal{P}}$ and reports $|\hat{\mathcal{P}}|$ as the count. As shown in Fig. 1, exemplar information is introduced at multiple levels of a frozen encoder before the conditioned features are aggregated for dense point prediction.

*Corresponding Author

¹<https://github.com/Shengjie-Guo/AgriCountDINO>

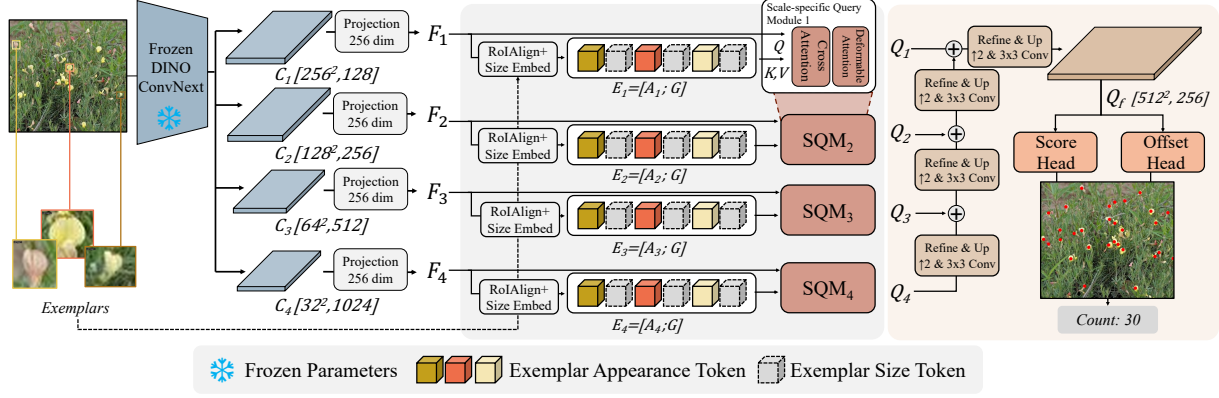


Fig. 1. Overview of AgriCountDINO. Exemplar appearance and size tokens condition frozen multiscale features through scale-specific query modules (SQMs). Progressive coarse-to-fine fusion produces a high-resolution representation for point scoring and offset prediction. Feature dimensions are shown for a 1024×1024 input.

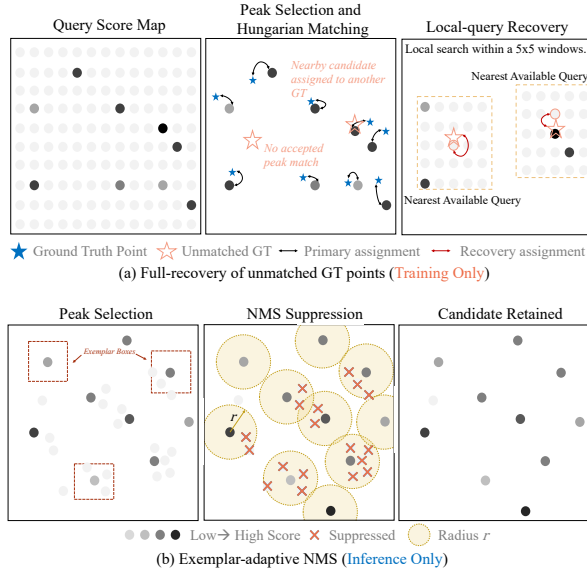


Fig. 2. Missed-object recovery and exemplar-adaptive point NMS. (a) During training, unmatched targets are assigned to unique unused queries in their local neighborhoods. (b) During inference, nearby candidate points are suppressed using a radius determined by the exemplar scale.

2.1. Multiscale Exemplar Conditioning

A frozen DINOv3-pretrained encoder [11, 12] extracts four feature maps. For a 1024×1024 input, C_1, C_2, C_3, C_4 have sizes $256^2 \times 128, 128^2 \times 256, 64^2 \times 512$, and $32^2 \times 1024$, respectively. A separate 1×1 convolution projects each map to 256 channels, producing $\{F_l\}_{l=1}^4$.

At each scale, RoIAlign [13] extracts an appearance token from each of the K exemplar boxes, while a shared lightweight MLP embeds each box’s normalized width and height into a size token. The K appearance tokens and K size tokens form $E_l \in \mathbb{R}^{2K \times 256}$. Inspired by previous work [14], a scale-specific query module (SQM) uses tokens from F_l as queries and E_l as keys and values. Cross-attention relates image locations to the exemplars’ appearance and

size cues, and single-scale deformable attention [15] refines the resulting features, yielding $\{Q_l\}_{l=1}^4$.

2.2. Progressive Dense Point Decoding

The conditioned maps are fused from coarse to fine by upsampling the fused feature from the previous level and adding it to the next finer map. Let $\mathcal{R}_l$ denote $2 \times$ bilinear upsampling followed by a 3×3 convolution and GELU. The decoder computes

$$\begin{aligned} Z_4 &= Q_4, \\ Z_l &= Q_l + \mathcal{R}_l(Z_{l+1}), \quad l = 3, 2, 1, \\ Q_f &= \mathcal{R}_f(Z_1). \end{aligned} \quad (1)$$

Here, Z_l is the fused feature at level l . For a 1024×1024 input, the final upsampling produces Q_f with 256 channels at a resolution of 512×512 .

Each position i in Q_f defines a feature vector q_i and a reference point r_i at the center of its grid cell in normalized image coordinates. Score and coordinate heads predict

$$s_i = h_s(q_i), \quad \hat{p}_i = \text{clip}(r_i + 2\sigma(h_p(q_i)) - 1, 0, 1). \quad (2)$$

Here, h_s produces a target score, while $2\sigma(h_p(q_i)) - 1$ gives a two-dimensional displacement; clip bounds the resulting point to the image. A score peak may lie outside the grid cell containing its target center, particularly in crowded regions. Allowing the predicted point to cross cell boundaries lets that query localize the target center.

2.3. Localization-Aware Learning and Inference

The dense decoder provides a scored point at every position in Q_f . To establish unambiguous supervision, 3×3 local score maxima above the image-wise median are matched one-to-one with annotated points. The matching favors confident predictions close to target centers. For a selected candidate k and annotation g_j , its cost is

$$C_{kj} = \lambda_p \|\hat{p}_k - g_j\|_1 - \lambda_s \sigma(s_k). \quad (3)$$

Here, $\hat{p}_k$ and g_j are normalized image coordinates, σ is the sigmoid function, and λ_p and λ_s balance localization and confidence. Matched pairs are retained only if their Euclidean distance in image pixels falls within an exemplar-dependent tolerance.

In crowded regions, closely spaced instances can produce overlapping responses on the score map, leaving fewer distinct peaks than annotated targets. Peak-based matching may therefore leave some targets without a positive query. Missed-object recovery (Fig. 2(a)) extends supervision to these targets by assigning them unused dense queries. Candidate reference positions r_i are searched within a 5×5 grid neighborhood of each unmatched annotation g_j , followed by a 7×7 search for any remaining targets. A joint one-to-one assignment minimizes the total reference-to-annotation ℓ_1 distance while reserving queries accepted by the initial matching. This gives nearby targets distinct training queries.

The initial matching and missed-object recovery define the positive queries for training. Accepted matches and recovered queries receive positive score labels; the remaining selected peaks receive negative labels. Their supervision is organized as

$$\mathcal{L} = \mathcal{L}_{\text{score}} + \lambda_{\text{loc}} \mathcal{L}_{\text{loc}} + \rho_t \left(\lambda_{\text{xy}} \mathcal{L}_{\text{rec}}^{\text{xy}} + \lambda_{\text{peak}} \mathcal{L}_{\text{rec}}^{\text{peak}} \right) + \alpha(\mathcal{E}) \mathcal{L}_{\text{aux}}. \quad (4)$$

The masked squared-error term $\mathcal{L}_{\text{score}}$ supervises selected peaks and recovered queries. The ℓ_1 term $\mathcal{L}_{\text{loc}}$ trains point coordinates on accepted initial matches. For recovered queries, $\mathcal{L}_{\text{rec}}^{\text{xy}}$ penalizes coordinate error, while $\mathcal{L}_{\text{rec}}^{\text{peak}}$ encourages each query to score above nearby unassigned queries in its 3×3 neighborhood. Accepted and recovered positives are excluded from this comparison, and both recovery terms place greater weight on closely spaced targets relative to the exemplar scale. Recovered queries enter $\mathcal{L}_{\text{score}}$ from the first epoch; ρ_t gradually introduces only their coordinate and peak losses over the first eight epochs. The auxiliary loss $\mathcal{L}_{\text{aux}}$ supervises predictions from Q_1 and is weighted by $\alpha(\mathcal{E})$ for small targets.

At inference, confident local score maxima yield candidate points, but several peaks may regress to the same target. Exemplar-adaptive point NMS (Fig. 2(b)) processes candidates in descending score order and suppresses points near each retained prediction. Its radius is based on $\frac{1}{2} \min(\bar{w}, \bar{h})$, where $\bar{w}$ and $\bar{h}$ are the mean exemplar width and height in image pixels. The retained points form $\hat{\mathcal{P}}$, whose cardinality gives the count.

3. EXPERIMENTS

3.1. Experimental Setup

3.2. Experimental Setup

Implementation details. AgriCountDINO uses a frozen DINOv3-pretrained ConvNeXt-B encoder [11, 12]. Images are resized to 1024×1024 , with up to three exemplar boxes used during training. The trainable parameters are optimized for 100 epochs with AdamW [18], a learning rate and weight decay of 10^{-4} , an effective batch size of 32, and BF16 mixed precision. The matching weights are $\lambda_p = 1$ and $\lambda_s = 2$; the loss weights are $\lambda_{\text{loc}} = 1$, $\lambda_{\text{xy}} = 0.5$, and $\lambda_{\text{peak}} = 0.05$. Recovery coordinate and peak losses are gradually introduced over the first eight epochs. The auxiliary weight is 0.3 when the smaller mean exemplar dimension is below 25 pixels, and zero otherwise.

Datasets and metrics. AgriCountDINO is trained on the 7,000 images in the official TPC-268 training split and evaluated in one-shot and three-shot settings. MAE, RMSE, and R^2 measure image-level counting accuracy. To assess the spatial distribution of predicted counts, GAME- L [19] sums absolute count errors over 4^L image regions; we report levels $L = 1, 2, 3$. This exposes regional errors that can cancel in the image-level count. Point-level precision and

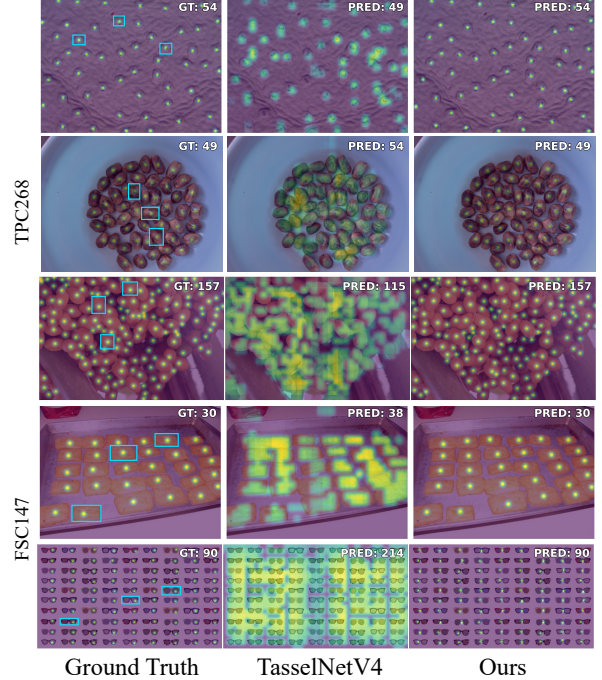


Fig. 3. Qualitative comparison with TasselNetV4. Exemplar boxes are shown in the ground truth, and dots denote target locations. The top three rows are from TPC-268; the bottom two show zero-shot transfer to FSC-147.

recall assess individual target localization. Zero-shot transfer is evaluated on the generic-object dataset FSC-147 without target-domain training or fine-tuning.

3.3. Counting Accuracy and Spatial Localization

With 8.40M trainable parameters, approximately one tenth of TasselNetV4’s, AgriCountDINO achieves the lowest TPC-268 validation MAE and RMSE in both exemplar settings (Table 1). In the three-shot setting, its MAE of 11.92 and RMSE of 28.55 reduce the corresponding errors of TasselNetV4 by 9.7% and 35.0%. With one exemplar, it obtains a validation MAE of 13.16. On the test split, the three-shot and one-shot MAEs are 21.76 and 22.03, compared with 22.95 and 22.20 for TasselNetV4.

To examine whether accurate counts are accompanied by accurate target placement, we also evaluate predicted point localization. In the three-shot setting, AgriCountDINO reduces GAME3 from 33.33 for TasselNetV4 to 25.37, a 23.9% reduction at the finest reported grid level (Table 2). Its predicted points achieve 0.719 precision and 0.729 recall. The examples in Fig. 3 show points aligned with individual targets, including closely spaced instances where TasselNetV4 produces broader counting responses.

Remaining errors on the validation and test sets may arise from both difficult target configurations and annotation quality. Dense or visually ambiguous instances can challenge point prediction, while Fig. 4 shows examples of apparent discrepancies between visible targets and point annotations. A prediction on a plausible but unannotated target may therefore increase the measured error.

Table 1. Comparison with representative state-of-the-art methods on TPC-268. Best and second-best results under each shot setting are shown in **bold** and underline, respectively.

Method	Shot	Validation			Test			Train. Params.↓
		MAE↓	RMSE↓	R^2 ↑	MAE↓	RMSE↓	R^2 ↑	
LOCA (ICCV’23)[16]	3	17.26	53.19	0.75	17.51	38.37	0.78	<u>11.37M</u>
CACViT (AAAI’24)[17]	3	16.63	<u>42.49</u>	0.82	22.04	<u>41.79</u>	0.73	99.77M
TasselNetV4 (ISPRS J.’26)[1]	3	<u>13.20</u>	43.93	<u>0.83</u>	22.95	51.36	0.67	84.67M
AgriCountDINO (Ours)	3	11.92	28.55	0.93	<u>21.76</u>	49.58	<u>0.74</u>	8.40M
LOCA (ICCV’23)[16]	1	17.19±0.31	48.14±2.19	0.80±0.02	21.47±0.29	42.36±0.72	0.73±0.01	<u>11.37M</u>
CACViT (AAAI’24)[17]	1	17.96±0.16	43.38±0.47	0.83±0.00	22.06±0.11	<u>42.97±0.81</u>	<u>0.71±0.01</u>	99.77M
TasselNetV4 (ISPRS J.’26)[1]	1	<u>13.49±0.02</u>	41.30±0.46	<u>0.85±0.00</u>	22.20±0.11	48.70±0.26	0.67±0.00	84.67M
AgriCountDINO (Ours)	1	13.16±0.07	33.16±0.66	0.90±0.00	<u>22.03±0.18</u>	50.08±0.46	0.70±0.00	8.40M

Table 2. Spatial counting and localization on TPC-268 validation. G1–G3 denote GAME1–GAME3.

Method	Shot	G1↓	G2↓	G3↓	Prec.↑	Rec.↑
TasselNetV4 [1]	3	17.71	22.75	33.33	–	–
TasselNetV4 [1]	1	19.14	23.86	33.93	–	–
AgriCountDINO	3	14.92	18.64	25.37	.719	.729
AgriCountDINO	1	15.90	19.72	27.17	.713	.709

Table 3. Three-shot in-domain and zero-shot TPC-268→FSC-147 results.

Method	FSC-147→FSC-147		TPC-268→FSC-147	
	MAE↓	RMSE↓	MAE↓	RMSE↓
LOCA [16]	<u>10.79</u>	<u>56.97</u>	<u>15.16</u>	109.15
CACViT [17]	9.13	48.96	17.88	82.57
TasselNetV4 [1]	16.16	110.56	19.17	116.24
AgriCountDINO	11.62	85.07	14.25	<u>106.88</u>

3.4. Zero-Shot Cross-Dataset Transfer

Trained only on TPC-268, AgriCountDINO transfers directly to unseen generic object categories in FSC-147 without target-domain training or fine-tuning. Table 3 reports a zero-shot MAE of 14.25, improving on LOCA’s 15.16 by 6.0%. The FSC-147 examples in Fig. 3 further show that the model localizes individual targets in these unseen categories. These results suggest that the model captures transferable semantic correspondences between exemplars and target objects, enabling counting and localization beyond the agricultural categories seen during training.

3.5. Ablation Study

The ablations in Table 4 reveal distinct roles for the three components. Replacing DINOv3 with supervised ConvNeXt-B causes the largest increase in MAE (11.92 to 17.64), suggesting that the DINOv3 representation provides a stronger basis for relating exemplars to diverse plant targets. Removing scale-wise conditioning worsens both MAE and GAME3, indicating that exemplar guidance across the feature hierarchy contributes to accurate global and regional counts. Without missed-object recovery, point recall falls from 0.729 to 0.692 and MAE increases, consistent with more targets

Table 4. Ablation study on the TPC-268 validation set under the 3-shot setting.

Configuration	MAE↓	G3↓	F1↑	Recall↑
Supervised ConvNeXt-B prior	17.64	32.19	0.668	0.646
w/o scale-wise exemplar conditioning	15.12	31.78	0.669	0.632
w/o missed-object recovery	12.71	26.27	0.723	0.692
Full model	11.92	25.37	0.723	0.729

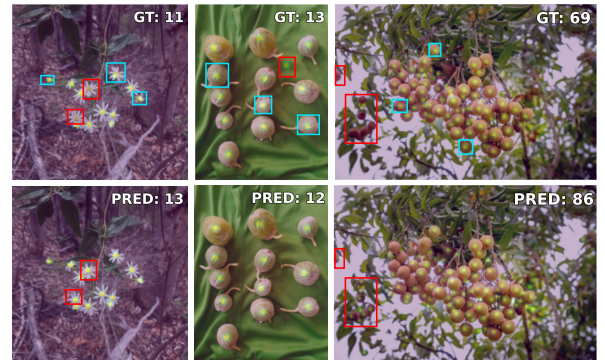


Fig. 4. Possible annotation discrepancies in TPC-268. Top: annotations and exemplar boxes; bottom: AgriCountDINO predictions. Red boxes highlight apparent discrepancies

being omitted after initial peak matching. F1 remains unchanged, so the clearest contribution of recovery is improved instance coverage.

4. CONCLUSIONS

We introduced AgriCountDINO, a parameter-efficient model for exemplar-guided counting and localization of plants and plant organs. It conditions frozen multiscale DINOv3 features on exemplar appearance and size, then predicts individual target points whose number gives the count. With 8.4M trainable parameters, AgriCountDINO achieves lower validation counting and spatial errors than TasselNetV4 on TPC-268. It also obtains the lowest zero-shot MAE among the compared methods on FSC-147 without target-domain fine-tuning. These results demonstrate AgriCountDINO’s ability to combine accurate counting, instance localization, and transfer beyond the agricultural categories used for training.

5. ACKNOWLEDGMENT

This work was co-funded by the European Union through the FutureData4EU project (Grant Agreement No. 101126733). The views expressed are those of the authors and do not necessarily reflect those of the European Union or REA. We acknowledge ISCRA for awarding access to the LEONARDO supercomputer, owned by the EuroHPC Joint Undertaking and hosted by CINECA (Italy).

6. REFERENCES

- [1] Xiaonan Hu, Xuebing Li, Jinyu Xu, Abdulkadir Duran Adan, Letian Zhou, Xuhui Zhu, Yanan Li, Wei Guo, Shouyang Liu, Wenzhong Liu, and Hao Lu, “Tasselnetv4: A vision foundation model for cross-scene, cross-scale, and cross-species plant counting,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 231, pp. 745–760, 2026.
- [2] Qingyu Song, Changan Wang, Zhengkai Jiang, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yang Wu, “Rethinking counting and localization in crowds: A purely point-based framework,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 3365–3374.
- [3] Thanh Nguyen, Chau Pham, Khoi Nguyen, and Minh Hoai, “Few-shot object counting and detection,” in *Computer Vision – ECCV 2022*, Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, Eds., Cham, 2022, pp. 348–365, Springer Nature Switzerland.
- [4] Jer Pelhan, Alan Lukežić, Vitjan Zavrtanik, and Matej Kristan, “Dave - a detect-and-verify paradigm for low-shot counting,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 23293–23302.
- [5] Jer Pelhan, Alan Lukežić, Vitjan Zavrtanik, and Matej Kristan, “A novel unified architecture for low-shot counting by detection and segmentation,” in *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, Eds. 2024, vol. 37, pp. 66260–66282, Curran Associates, Inc.
- [6] Hao Lu, Zhiguo Cao, Yang Xiao, Bohan Zhuang, and Chunhua Shen, “TasselNet: Counting maize tassels in the wild via local counts regression network,” *Plant Methods*, vol. 13, no. 1, pp. 79, 2017.
- [7] Jinyu Xu, Tianqi Hu, Xiaonan Hu, Letian Zhou, Songliang Cao, Meng Zhang, and Hao Lu, “Plant taxonomy meets plant counting: A fine-grained, taxonomic dataset for counting hundreds of plant species,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026, pp. 167–177.
- [8] Min Shi, Hao Lu, Chen Feng, Chengxin Liu, and Zhiguo Cao, “Represent, compare, and learn: A similarity-aware framework for class-agnostic counting,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 9529–9538.
- [9] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas, “Self-supervised learning from images with a joint-embedding predictive architecture,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 15619–15629.
- [10] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin, “Emerging properties in self-supervised vision transformers,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 9650–9660.
- [11] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Theo Moutakanni, Leo Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski, “Dinov3,” *arXiv preprint*, 2025.
- [12] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie, “A convnet for the 2020s,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11976–11986.
- [13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, “Mask R-CNN,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2961–2969.
- [14] Jer Pelhan, Alan Lukežić, and Matej Kristan, “Generalized-scale object counting with gradual query aggregation,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026, vol. 40, pp. 8314–8321.
- [15] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai, “Deformable DETR: Deformable transformers for end-to-end object detection,” in *International Conference on Learning Representations (ICLR)*, 2021.
- [16] Nikola Đukić, Alan Lukežić, Vitjan Zavrtanik, and Matej Kristan, “A low-shot object counting network with iterative prototype adaptation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 18872–18881.
- [17] Zhicheng Wang, Liwen Xiao, Zhiguo Cao, and Hao Lu, “Vision transformer off-the-shelf: A surprising baseline for few-shot class-agnostic counting,” *arXiv preprint arXiv:2305.04440*, 2023.
- [18] Ilya Loshchilov and Frank Hutter, “Decoupled weight decay regularization,” in *International Conference on Learning Representations*, 2019.
- [19] Ricardo Guerrero-Gómez-Olmedo, Beatriz Torre-Jiménez, Roberto López-Sastre, Saturnino Maldonado-Bascón, and Daniel Oñoro-Rubio, “Extremely overlapping vehicle counting,” in *Pattern Recognition and Image Analysis (IbPRIA)*. 2015, pp. 423–431, Springer.