CountingDINO
A Training-free Pipeline for Class-Agnostic Counting using Unsupervised Backbones

Giacomo Pacini *[1,2], Lorenzo Bianchi *[1,2],
Luca Ciampi[1], Nicola Messina[1], Giuseppe Amato[1], and Fabrizio Falchi[1]

[1] CNR-ISTI, Pisa, Italy - {name.surname}@isti.cnr.it
[2] University of Pisa, Italy - {name.surname}@phd.unipi.it
*Equal contribution

Abstract. Class-agnostic counting (CAC) aims to estimate the number of objects in images without being restricted to predefined categories. However, while current exemplar-based CAC methods offer flexibility at inference time, they still rely heavily on labeled data for training, which limits scalability and generalization to many downstream use cases. In this paper, we introduce CountingDINO, the first *training-free* exemplar-based CAC framework that exploits a *fully unsupervised* feature extractor. Specifically, our approach employs self-supervised vision-only backbones to extract object-aware features, and it eliminates the need for annotated data throughout the entire proposed pipeline. At inference time, we extract latent object prototypes via ROI-Align from DINO features and use them as convolutional kernels to generate similarity maps. These are then transformed into density maps through a simple yet effective normalization scheme. We evaluate our approach on the FSC-147 benchmark, where we outperform a baseline under the same label-free setting. Our method also achieves competitive – and in some cases superior – results compared to training-free approaches relying on supervised backbones, as well as several fully supervised state-of-the-art methods. This demonstrates that training-free CAC can be both scalable and competitive. Website: https://lorebianchi98.github.io/CountingDINO/.

1 Introduction

Class-agnostic counting (CAC) aims to count instances of arbitrary object classes beyond those encountered during training [25]. This recent paradigm addresses the inherent limitations of traditional class-specific counting approaches, which rely on models trained for predefined object types, such as vehicles [1, 33], people [4, 20], cells [8, 32], or animals [2, 28]. Unlike these methods, CAC allows users to dynamically define target categories during inference, removing the need to retrain deep learning-based networks with class-specific annotated datasets.

However, CAC still relies heavily on human annotations. First, most existing approaches extract image features using backbones pre-trained on large-scale labeled datasets such as ImageNet [16]. Second, they require supervised training

on annotated datasets containing thousands of objects across hundreds of categories. In particular, the most widely adopted CAC paradigm – the exemplar-based approach – typically involves two types of supervision: dot annotations and a small number of bounding boxes. Dot annotations, placed on object centroids, are used to generate density maps that serve as training targets for a regression network learning to map image features to object density estimates [19]. Bounding boxes, on the other hand, are used to localize a small set of visual prototypes, known as *exemplars*, which represent the object category to be counted [25]. Particularly limiting is the requirement for thousands of dot annotations, which poses a major obstacle to the creation of large-scale datasets. As a result, only a few datasets are available, and even these often exhibit notable limitations [9].

In this paper, we introduce CountingDINO, the first *training-free* exemplar-based CAC framework that eliminates the need for labeled data at any stage. Our image feature extraction backbone builds on DINO [6, 11, 23], which is trained in a self-supervised fashion. At inference time, we extract exemplar features by applying ROI-Align [15] to user-provided bounding boxes using the same DINO-computed image feature maps. These exemplar features are then used as depth-wise convolutional kernels over the image features, generating similarity maps that highlight regions matching the exemplars. We average these similarity maps across exemplars to produce a global, informative response, which is then converted into a density map through a simple yet effective normalization scheme. Final object counts are computed by integrating the density map. Furthermore, to address the limitations of the spatial resolution of DINO with small objects, we partition the image into non-overlapping quadrants, apply feature extraction independently, and aggregate the resulting maps. We validate our method on the gold-standard CAC benchmark FSC-147 [25], outperforming a baseline under the same label-free setting. We also compare against training-free methods with supervised backbones and fully supervised state-of-the-art approaches, showing that our method remains competitive despite using no supervision.

To summarize, we propose the following contributions:

- We introduce CountingDINO, the first *training-free* class-agnostic counting framework that utilizes self-supervised backbones to extract object-aware features, removing any dependency on annotated data.
- On the FSC-147 benchmark, we show that CountingDINO outperforms a baseline under the same label-free setting, and achieves performance comparable to – and sometimes surpassing – recent training-free methods with supervised backbones, as well as fully supervised approaches.
- We conduct ablation studies to assess the contribution of each core component in our model.

2 Related Works

2.1 Exemplar-based Class-agnostic Counting

Counting category-specific objects is a longstanding task in computer vision with broad real-world applications – e.g., counting of people [4, 20], vehicles [1, 33], in-

sects [5,10], or biological structures [8,32]. Among existing counting approaches, density map regression — estimating counts via feature-to-density mapping — has proven effective in crowded scenes [2, 19], outperforming detection-based techniques [1, 28]. However, their reliance on category-specific annotations and training limits scalability and generalization.

Class-agnostic counting (CAC) has recently shifted the paradigm of object counting toward open-world contexts, enabling models to handle arbitrary object categories not encountered during training. In this setting, users are no longer restricted to predefined categories and can instead specify novel object classes at inference time by providing visual exemplars — typically in the form of three bounding boxes enclosing visual prototypes within the same input image [25]. The seminal FamNet [25] combines multi-scale feature extraction with exemplar-image correlation via inner product operations for density map prediction. RCAC [13] follows a similar architecture but introduces a feature augmentation module that synthesizes exemplars with varied colors, shapes, and scales to enhance diversity. BMNet [26] highlights the limitations of inner product-based correlation and instead proposes a learnable bilinear similarity loss, inspired by metric learning, for better matching. LOCA [29] introduces an object prototype extraction module that iteratively refines exemplar features via cross-attention, while PseCo [17] relies on the popular SAM for instance segmentation [18]. CACViT [31] leverages a single pre-trained Vision Transformer, using its attention mechanism for both feature extraction and matching. DAVE [24] proposes a two-stage detect-and-verify framework: first identifying candidate boxes via density maps and then verifying them using exemplar-based clustering. In contrast to these fully supervised methods, TFPOC [27] and OmniCount [21] propose training-free pipelines that rely on SAM without point-level supervision – though SAM itself is trained with labels, making their approach not entirely label-free. A recent survey of CAC methods is available in [7].

2.2 Unsupervised Vision Backbones

Recent advances in unsupervised – especially self-supervised – learning have yielded models whose learned representations are highly effective for classification and localization. The DINO family [6, 11, 23], based on vision transformers [12] and trained via self-distillation, demonstrates that patch-level features learned without labels can capture rich semantic information, outperforming CLIP-based models, which rely on weak supervision from image-text pairs. Alongside DINO, masked autoencoders (MAE) [14] provide another powerful self-supervised approach by training transformers to reconstruct masked image patches, fostering global visual understanding.

In this work, we propose a *training-free* exemplar-based CAC framework that leverages *fully unsupervised* vision backbones for extracting features from both images and exemplars. Our pipeline requires no labeled data at any stage, neither for training the image feature extractor nor for generating the density maps.

3 Method

Given an image and a small set of exemplar bounding boxes from the same image — each containing a visual prototype of the object we want to count — we aim to estimate the total number of instances of that object class present in the image, without relying on human-labeled data at any stage. To this end, we leverage the object understanding capabilities of DINO [6,11,23], a self-supervised vision-only backbone, and proceeding as described in the following sections. We also report a detailed schema in Fig. 1.

3.1 DINO-based Feature Extraction

Let $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ be an input image, and let ψ_{v} denote a DINO-based self-supervised visual backbone. Passing the image through ψ_{v} yields a dense feature map: $\mathbf{V} = \psi_{\text{v}}(\mathbf{I}) \in \mathbb{R}^{L \times V \times D}$, where $L = \frac{H}{P}$ and $V = \frac{W}{P}$ represent the spatial dimensions of the feature map, with P being the patch size and D the dimensionality of the feature embeddings. Each location in $\mathbf{V}$ encodes a feature vector corresponding to a specific image patch.

We also extract the representation from all the available exemplars. Each exemplar is specified by a bounding box in the original image space, defined by its top-left and bottom-right coordinates as $\mathbf{b} = (x_1, y_1, x_2, y_2)$, where $x_1, y_1, x_2, y_2 \in \mathbb{R}$. We simply perform ROI-Align on $\mathbf{V}$ using the bounding box $\mathbf{b}$, by carefully downsampling the coordinates of $\mathbf{b}$ to $\tilde{\mathbf{b}} = (\lfloor Lx_1 \rfloor, \lfloor Vy_1 \rfloor, \lceil Lx_2 \rceil, \lceil Vy_2 \rceil)$. The output of this operation is a feature $\mathbf{R}^{\tilde{\mathbf{b}}}$ that captures the visual appearance of the exemplar $\tilde{\mathbf{b}}$ in the embedding space.

To refine the exemplar representation, we apply a soft spatial prior in the form of an elliptical weighting mask centered within the bounding box. This mask attenuates peripheral regions and emphasizes the center under the assumption that exemplar objects are typically centered within their boxes. Formally, we define a mask $\mathbf{M}^{\tilde{\mathbf{b}}} \in \mathbb{R}^{w(\tilde{\mathbf{b}}) \times h(\tilde{\mathbf{b}})}$, where each entry $m_{ij}^{\tilde{\mathbf{b}}}$ indicates the proportion of the corresponding feature cell in the bounding box $\tilde{\mathbf{b}}$ that lies within an ellipse centered in it. The impact of this spatial prior and the underlying assumption is further analyzed through ablation experiments in Sec. 4.4.

3.2 Similarity Map Generation

In this stage we aim at creating a 2D similarity map where each value reflects how closely the local region in the image feature map matches the exemplar in both appearance and structure. To obtain this, we use each pooled exemplar feature $\mathbf{R}^{\tilde{\mathbf{b}}}$ as a convolutional kernel on the map $\mathbf{V}$, i.e., $\mathbf{S}^{\tilde{\mathbf{b}}} = \text{Conv2D}(\mathbf{V}, \mathbf{R}^{\tilde{\mathbf{b}}}) \in \mathbb{R}^{\bar{L} \times \bar{V}}$, where $\bar{L}$ and $\bar{V}$ are the spatial dimensions of the output map.

We repeat this process for each of the N exemplar bounding boxes $\{\tilde{\mathbf{b}}_i\}_{i=1}^{N}$, resulting in N similarity maps $\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_N$. Finally, we average the aligned maps to obtain a single aggregated similarity map $\mathbf{S} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{S}_i$, which reflects the overall likelihood of object occurrence across all exemplars.

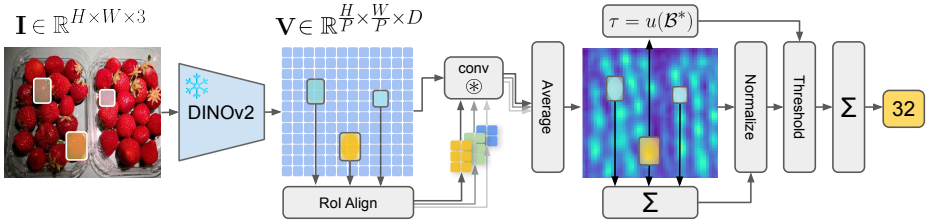

Fig. 1: Overview of CountingDINO. Given an image I and N exemplar boxes, we extract features using the DINO-based visual backbone and apply each exemplar as a convolutional kernel over the image feature map to obtain similarity maps. These are aggregated, normalized into a density map using spatial priors, and thresholded before integration to produce the final count.

3.3 Density Map Normalization

The aggregated similarity map $\mathbf{S}$ well highlights regions likely to contain object instances. However, we seek a similarity response that integrates to 1 over each object instance, effectively transforming the similarity map into a density map from which the total count can be estimated via integration.

In exemplar-based counting, we can leverage the N exemplar bounding boxes provided as input to properly estimate a normalization factor. In fact, we know that the area occupied by the N exemplars should count up to exactly N. To this aim, we reuse the elliptical masks $\{\mathbf{M}^{\tilde{\mathbf{b}}_i}\}_{i=1}^{N}$ introduced in Sec. 3.1 and compute the normalization factor z as follows:

$$z = \frac{1}{N} \sum_{(x,y) \in (\bar{L} \times \bar{V})} \mathbf{M}_{xy} \odot \text{minmax}(\mathbf{S})_{xy}, \tag{1}$$

where the global weighting mask $\mathbf{M}$ is obtained by properly rescaling the masks $\{\mathbf{M}^{\tilde{\mathbf{b}}_i}\}_{i=1}^{N}$ and accumulating them into a $\bar{L} \times \bar{V}$ map initialized with all zeros. Notice that, to ensure the normalization relies only on positive values, we first apply minmax normalization to $\mathbf{S}$, rescaling it to the range $[0, 1]$.

The final normalized map is obtained by dividing the minmax-ed activation map by the computed scalar factor z:

$$\hat{\mathbf{S}} = \frac{\text{minmax}(\mathbf{S})}{z} \tag{2}$$

This ensures the total response across the N exemplar regions sums to N, making the response per object approximate a unit mass. Consequently, the normalized map $\hat{\mathbf{S}}$ serves as a density estimate, with the total count obtained by integrating it over the spatial dimensions.

3.4 Thresholding density map contributions

The similarity map highlights object regions but also includes low activations in the background, which can cause overcounting during integration. To address this, we apply a thresholding step to suppress low-activation areas.

In order to find a reasonable threshold, we notice that the normalization criteria detailed in Eqs. 1 and 2 induce a mean per-patch unit count, estimated on the set $\mathcal{B}$ of exemplars, of $u(\mathcal{B}) = \frac{|\mathcal{B}|}{\sum_{\mathbf{b}\in\mathcal{B}} \text{area}(\mathbf{b})}$ — where area$(\cdot)$ is a function returning the area, expressed in the number of patches, for the given bounding box. The resulting term $u(\mathcal{B})$ is the mean unit count value that a patch carries if it is part of a relevant object. This means that it is likely that patches carrying less value than $u(\mathcal{B})$ pertain to a non-relevant object and may be filtered out.

However, notice that $u(\mathcal{B})$ assumes its minimum value when fed with the set $\mathcal{B}^* = \{\arg\max_{\mathbf{b}\in\mathcal{B}} \text{area}(\mathbf{b})\}$ — i.e. when computed on the set composed solely of the largest exemplar bounding box. In cases different from this, we are also inevitably filtering out contributions from the other exemplars bounding boxes that we know are valid. For this reason, we set the threshold to be exactly the unit count computed on the largest region, i.e., $\tau = u(\mathcal{B}^*)$.

This process ensures that only the relevant activations are considered, improving the accuracy of the object count estimate.

3.5 Increasing Spatial Resolution

The above-presented method generates consistent results when the spatial resolution of DINO is enough to capture no more than an object per visual patch, which happens when the objects are not too small. Therefore, instead of processing the entire image at once, we divide the input image into four non-overlapping and evenly-shaped quadrants and independently process each sub-image with the backbone ψ_{v}. In general, we divide the image into 4^k different quadrants, where $k \in \mathbb{N}$ defines the resolution level — i.e., how many times we recursively partition each quadrant in its four sub-quadrants.

After extracting features from each quadrant, we spatially reassemble the four feature maps into a single unified feature tensor by stitching them according to their original spatial layout. This yields a higher-resolution feature map $\mathbf{V} \in \mathbb{R}^{2^k L \times 2^k V \times D}$, effectively increasing the granularity of the representation and enabling more precise localization of small objects.

This modification integrates seamlessly into the pipeline, as all subsequent steps — ROI-Align, similarity computation, and normalization — remain unchanged. We found $k = 2$ to work well in the presented domain, although we experiment with other values in Sec. 4.4.

4 Experimental Evaluation

4.1 Datasets and Metrics

We evaluate our method on the FSC-147 dataset [25], which consists of 6,135 images belonging to 147 object classes, each accompanied by three exemplar bounding boxes. We evaluate counting performance using two standard metrics: Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) [2, 19, 25], defined as $MAE = \frac{1}{n}\sum_{i=1}^{n} |c_i - \hat{c}_i|$ and $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(c_i - \hat{c}_i)^2}$, where c_i and $\hat{c}_i$ are the ground truth and predicted counts for the i-th image.

4.2 Baseline

Since CountingDINO is the first training-free methodology for exemplar-based CAC using an unsupervised backbone, we design a comparison baseline by adapting CutLER [30] – an unsupervised object detector that leverages pseudo-labels derived from DINO features. As CutLER is not designed for exemplar-based matching, we adapt it to our setting as follows. We first apply CutLER to the target image to obtain a set of detected object bounding boxes. To filter these detections and retain only those corresponding to the exemplar object category, we extract visual features using DINO – specifically, DINO ViT-B/8, the same backbone used during CutLER's training. For each exemplar bounding box, we apply ROI pooling to obtain its feature representation, and average these representations to form a prototype feature vector representing the exemplar class. We then extract features from the detected bounding boxes and compute their similarity to the prototype vector. Detections with a similarity score above a threshold (set to 0.5) are retained, and their count provides the final prediction.

4.3 Comparison with SOTA

We quantitatively compare our approach against the adapted CutLER baseline described above to ensure a fair comparison under the same label-free setting as our method. Additionally, we evaluate our method against two SOTA training-free CAC approaches – TFPOC [27] and OmniCount [21] – which, unlike ours, rely on a supervised backbone. Finally, we benchmark our method against several fully supervised SOTA CAC methods, including FamNet [25], RCAC [13], BMNet [26], LOCA [29], PseCo [17], CACViT [31], and DAVE [24]. For our approach, we implement two variants using different DINO backbones: DINO [6] ViT-B/8 and DINOv2 [23] ViT-L/14 with registers [11].

Quantitative results on the FSC-147 validation and test splits are reported in Tab. 1, while qualitative examples are shown in Fig. 2. Despite being entirely training-free and not relying on human annotations at any stage, our approach – particularly when using DINOv2 features – achieves competitive performance. Notably, we outperform the CutLER-based baseline, the only other method operating under the same label-free setting, by a significant margin across all evaluation metrics on both validation and test splits. When compared to TFPOC and OmniCount – two SOTA training-free methods that benefit from a supervised backbone for feature extraction – our method achieves substantially lower RMSE, highlighting its robustness in challenging scenarios with dense object distributions, occlusions, and overlaps. We also report competitive results in terms of MAE. Finally, it is worth emphasizing that our method surpasses several fully supervised CAC methods in terms of RMSE, including FamNet, RCAC, BMNet, and PseCo. This is particularly remarkable given that these methods leverage supervision not only for feature extraction but also for density map regression, relying on costly point-level annotations. These results demonstrate that our approach, despite being supervision-free, can compete with and even outperform fully supervised alternatives in certain settings.

Method	Unsup.	Training-Free	Validation MAE ↓	Validation RMSE ↓	Test MAE ↓	Test RMSE ↓
FamNet (CVPR '21) [25]	✗	✗	23.75	69.07$^{\uparrow}$	22.08$^{\uparrow}$	99.54$^{\uparrow}$
RCAC (ECCV '22) [13]	✗	✗	20.54	60.78$^{\uparrow}$	20.21	81.86$^{\uparrow}$
BMNet+ (CVPR '22) [26]	✗	✗	15.74	58.53$^{\uparrow}$	14.62	91.83$^{\uparrow}$
LOCA (ICCV '23) [29]	✗	✗	10.24	32.56	10.79	56.97
PseCo (CVPR '24) [17]	✗	✗	15.31	68.34$^{\uparrow}$	13.05	112.86$^{\uparrow}$
CACViT (AAAI '24) [31]	✗	✗	10.63	37.95	9.13	48.96
DAVE (CVPR '24) [24]	✗	✗	<u>8.91</u>	<u>28.08</u>	<u>8.66</u>	<u>32.36</u>
TFPOC (WACV '24) [27]	✗	✓	-	-	19.95	132.16$^{\uparrow}$
OmniCount (AAAI '25) [21]	✗	✓	-	-	<u>18.63</u>	<u>112.98</u>$^{\uparrow}$
CutLER Baseline	✓	✓	54.18	135.29	56.44	158.01
CountingDINO (DINO)	✓	✓	42.29	87.87	30.05	90.3
CountingDINO (DINOv2)	✓	✓	**25.48**	**57.38**	**20.93**	**71.37**

Table 1: SOTA comparison on FSC-147 (val/test). Methods are grouped as unsupervised or training-free. Best per category is <u>underlined</u>; best training-free method with an unsupervised backbone is in **bold**; $^{\uparrow}$marks results of methods in advantaged (supervised or non-training-free) categories performing worse than CountingDINO.

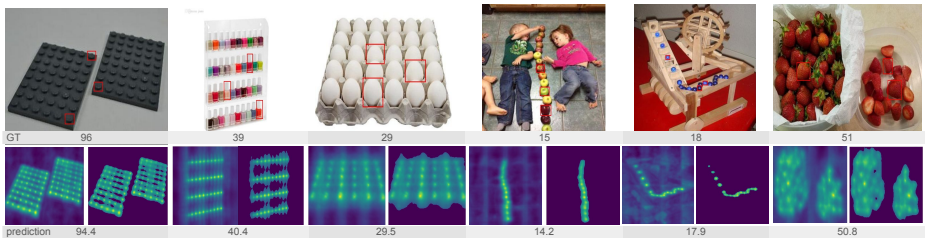

Fig. 2: Qualitative results. Samples using DINOv2 ViT L/14 Reg. as backbone. Below the images we report density maps before and after background thresholding.

4.4 Ablation Studies

Effect of the Elliptical Assumption. As introduced in Sec. 3, we assume that objects are more likely to appear near the center of exemplar bounding boxes and less likely in the corners. Section *A* of Tab. 2 shows that incorporating an elliptical weighting mask during feature extraction significantly improves performance. This spatial prior enhances object localization by emphasizing central regions and down-weighting peripheral areas, effectively suppressing background noise. It also benefits the normalization process by concentrating the density around the object and reducing the impact of irrelevant regions.

Effect of the Density Map Thresholding. Thresholding the similarity-based density map (Sec. 3.4) helps suppress spurious background activations. Section *B* of Tab. 2 shows that this step yields clear improvements in both MAE and

RMSE. As illustrated in Fig. 2, it also refines the density maps by reducing background noise and focusing responses on true object locations.

	Validation		Test	
	MAE ↓	RMSE ↓	MAE ↓	RMSE ↓
A) Effect of Ellipse assumption				
w/o ellipse	32.32	86.83	29.73	102.38
w/ ellipse	**25.48**	**57.38**	**20.93**	**71.37**
B) Effect of thresholding				
w/o thresholding	59.25	135.72	36.81	88.29
w/ thresholding	**25.48**	**57.38**	**20.93**	**71.37**
C) Effect of changing the resolution				
Resolution 1×	32.76	96.74	28.23	118.9
Resolution 2×	25.56	66.33	23.06	90.7
Resolution 4×	**25.48**	**57.38**	**20.93**	**71.37**
D) Effect of changing the backbone				
CLIP ViT B/16	139.48	383.41	72.34	196.49
CLIP ViT L/14	155.09	414.08	80.63	205.3
MAE ViT B/16	121.52	327.5	57.87	171.04
MAE ViT L/16	121.82	336.78	57.92	167.94
DINO ResNet50	66.18	141.87	53.87	126.46
DINO ViT S/8	115.92	280.36	55.64	158.95
DINO ViT B/8	42.29	87.87	30.05	90.3
DINOv2 ViT S/14	26.86	**57.05**	22.31	79.04
DINOv2 ViT B/14	26.0	57.26	21.39	80.55
DINOv2 ViT L/14	**24.97**	57.5	21.07	77.14
DINOv2 ViT S/14 reg.	26.68	57.29	**20.89**	76.02
DINOv2 ViT B/14 reg.	30.45	64.79	23.84	79.7
DINOv2 ViT L/14 reg.	25.48	57.38	20.93	**71.37**
E) Effect of changing the number of exemplars				
1 exemplar	39.87	113.70	37.73	131.71
2 exemplars	29.18	80.68	23.12	72.80
3 exemplars	**25.48**	**57.38**	**20.93**	**71.37**

Table 2: Ablation study. Impact of various components in our method. Unless specified otherwise, experiments use the DINOv2 ViT L/14 backbone with registers.

Effect of Different Resolutions. As shown in Section *C* of Tab. 2, increasing input resolution via image partitioning consistently improves performance. We evaluate three levels — k=0 (1×), k=1 (2×), and k=2 (4×)Gains are especially notable in RMSE, indicating that higher resolution helps disambiguate small, densely packed objects that may otherwise be lost at coarser scales.

Effect of the visual backbone. Beyond DINO and DINOv2, we evaluated alternative backbones, including MAE (unsupervised) and CLIP (trained with web-scale image-text supervision). As shown in Section *D* of Tab. 2, performance generally improves with backbone size within each family. DINOv2 consistently outperforms DINO, especially in smaller variants, highlighting its robustness. Registers further boost the DINOv2 model by reducing artifacts that hinder accurate counting. MAE underperforms DINO-based models due to less semantically rich patch features [3], while CLIP performs worst, confirming prior findings [3, 22] that its features are suboptimal for fine-grained object-centric tasks.

Effect of Varying the Number of Exemplars. We evaluate the robustness of our method under reduced exemplar counts. As shown in Section *E* of Tab. 2, increasing the number of exemplars improves counting accuracy, with the largest gain observed from 1 to 2 exemplars. This highlights the benefit of feature diversity and the ability of multiple prototypes to better capture appearance variations.

5 Conclusions

In this paper, we introduced the first fully unsupervised and training-free exemplar-based framework for class-agnostic counting. By leveraging self-supervised features from DINO, our method addresses a key limitation of existing CAC approaches that depend on costly human annotations and are biased to pay more attention to the labeled categories. Our approach surpasses all unsupervised and weakly supervised baselines on the FSC-147 benchmark, achieving performance competitive with supervised methods. These results validate the effectiveness of self-supervised representations in counting tasks, paving the way for scalable, annotation-free solutions in open-world scenarios. Future work will refine exemplar quality to further reduce counting noise and extend the method to prompt-based CAC tasks.

6 Acknowledgements

This work was partially funded by: Spoke 8, Tuscany Health Ecosystem (THE) Project (CUP B83C22003930001), funded by the National Recovery and Resilience Plan (NRRP), within the NextGeneration Europe (NGEU) Program; Horizon Europe Research & Innovation Programme under Grant agreement N. 101092612 (Social and hUman ceNtered XR - SUN project); PNRR - M4C2 - Investimento 1.3, Partenariato Esteso PE00000013 - "FAIR - Future Artificial Intelligence Research" - Spoke 1 "Human-centered AI", funded by European Union - NextGenerationEU; ITSERR - ITalian Strengthening of the Esfri Ri Resilience (CUP B53C22001770006), also funded by the European Union via NextGenerationEU.

References

1. Amato, G., Ciampi, L., Falchi, F., Gennaro, C.: Counting vehicles with deep learning in onboard UAV imagery. In: 2019 IEEE Symposium on Computers and Communications, ISCC 2019, Barcelona, Spain, June 29 - July 3, 2019. pp. 1–6. IEEE (2019). https://doi.org/10.1109/ISCC47284.2019.8969620 1, 2, 3
2. Arteta, C., Lempitsky, V.S., Zisserman, A.: Counting in the wild. In: ECCV. vol. 9911, pp. 483–498. Springer (2016). https://doi.org/10.1007/978-3-319-46478-7_30 1, 3, 6
3. Barsellotti, L., Bianchi, L., Messina, N., Carrara, F., Cornia, M., Baraldi, L., Falchi, F., Cucchiara, R.: Talking to dino: Bridging self-supervised vision backbones with language for open-vocabulary segmentation. arXiv preprint arXiv:2411.19331 (2024) 10
4. Benedetto, M.D., Carrara, F., Ciampi, L., Falchi, F., Gennaro, C., Amato, G.: An embedded toolset for human activity monitoring in critical environments. Expert Syst. Appl. **199**, 117125 (2022). https://doi.org/10.1016/J.ESWA.2022.117125 1, 2
5. Bereciartua-Perez, A., Gómez, L., Picón, A., Navarra-Mestre, R., Klukas, C., Eggers, T.: Insect counting through deep learning-based density maps estimation. Comput. Electron. Agric. **197**, 106933 (2022). https://doi.org/10.1016/J.COMPAG.2022.106933 3
6. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging Properties in Self-Supervised Vision Transformers. In: ICCV (2021) 2, 3, 4, 7
7. Ciampi, L., Azmoudeh, A., Akbaba, E.E., Saritas, E., Yazici, Z.A., Ekenel, H.K., Amato, G., Falchi, F.: A survey on class-agnostic counting: Advancements from reference-based to open-world text-guided approaches. CoRR **abs/2501.19184** (2025). https://doi.org/10.48550/ARXIV.2501.19184 3
8. Ciampi, L., Carrara, F., Totaro, V., Mazziotti, R., Lupori, L., Santiago, C., Amato, G., Pizzorusso, T., Gennaro, C.: Learning to count biological structures with raters' uncertainty. Medical Image Anal. **80**, 102500 (2022). https://doi.org/10.1016/J.MEDIA.2022.102500 1, 3
9. Ciampi, L., Messina, N., Pierucci, M., Amato, G., Avvenuti, M., Falchi, F.: Mind the prompt: A novel benchmark for prompt-based class-agnostic counting. In: WACV. pp. 7970–7979 (2025). https://doi.org/10.1109/WACV61041.2025.00774 2
10. Ciampi, L., Zeni, V., Incrocci, L., Canale, A., Benelli, G., Falchi, F., Amato, G., Chessa, S.: A deep learning-based pipeline for whitefly pest abundance estimation on chromotropic sticky traps. Ecol. Informatics **78**, 102384 (2023). https://doi.org/10.1016/J.ECOINF.2023.102384 3
11. Darcet, T., Oquab, M., Mairal, J., Bojanowski, P.: Vision Transformers Need Registers. In: ICLR (2024) 2, 3, 4, 7
12. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In: ICLR (2021) 3
13. Gong, S., Zhang, S., Yang, J., Dai, D., Schiele, B.: Class-agnostic object counting robust to intraclass diversity. In: ECCV. vol. 13693, pp. 388–403. Springer (2022). https://doi.org/10.1007/978-3-031-19827-4_23 3, 7, 8

14. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: CVPR. pp. 16000–16009 (2022) 3
15. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. IEEE TPAMI **42**(2), 386–397 (2020). https://doi.org/10.1109/TPAMI.2018.2844175 2
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778. IEEE (2016). https://doi.org/10.1109/CVPR.2016.90 1
17. Huang, Z., Dai, M., Zhang, Y., Zhang, J., Shan, H.: Point, segment and count: A generalized framework for object counting. In: CVPR. pp. 17067–17076. IEEE (2024). https://doi.org/10.1109/CVPR52733.2024.01615 3, 7, 8
18. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W., Dollár, P., Girshick, R.B.: Segment anything. In: ICCV. pp. 3992–4003. IEEE (2023). https://doi.org/10.1109/ICCV51070.2023.00371 3
19. Lempitsky, V.S., Zisserman, A.: Learning to count objects in images. In: NeurIPS. pp. 1324–1332. Curran Associates, Inc. (2010) 2, 3, 6
20. Liu, W., Salzmann, M., Fua, P.: Context-aware crowd counting. In: CVPR. pp. 5099–5108. Computer Vision Foundation / IEEE (2019). https://doi.org/10.1109/CVPR.2019.00524 1, 2
21. Mondal, A., Nag, S., Zhu, X., Dutta, A.: Omnicount: Multi-label object counting with semantic-geometric priors (2025 (Accepted - To appear)) 3, 7, 8
22. Mukhoti, J., Lin, T.Y., Poursaeed, O., Wang, R., Shah, A., Torr, P.H., Lim, S.N.: Open vocabulary semantic segmentation with patch aligned contrastive learning. In: CVPR. pp. 19413–19423 (2023) 10
23. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: DINOv2: Learning Robust Visual Features without Supervision. arXiv preprint arXiv:2304.07193 (2023) 2, 3, 4, 7
24. Pelhan, J., Lukezic, A., Zavrtanik, V., Kristan, M.: DAVE - A detect-and-verify paradigm for low-shot counting. In: CVPR. pp. 23293–23302. IEEE (2024). https://doi.org/10.1109/CVPR52733.2024.02198 3, 7, 8
25. Ranjan, V., Sharma, U., Nguyen, T., Hoai, M.: Learning to count everything. In: CVPR. pp. 3394–3403. Computer Vision Foundation / IEEE (2021). https://doi.org/10.1109/CVPR46437.2021.00340 1, 2, 3, 6, 7, 8
26. Shi, M., Lu, H., Feng, C., Liu, C., Cao, Z.: Represent, compare, and learn: A similarity-aware framework for class-agnostic counting. In: CVPR. pp. 9519–9528. IEEE (2022). https://doi.org/10.1109/CVPR52688.2022.00931 3, 7, 8
27. Shi, Z., Sun, Y., Zhang, M.: Training-free object counting with prompts. In: WACV. pp. 322–330. IEEE (2024). https://doi.org/10.1109/WACV57701.2024.00039 3, 7, 8
28. Tian, M., Guo, H., Chen, H., Wang, Q., Long, C., Ma, Y.: Automated pig counting using deep learning. Comput. Electron. Agric. **163** (2019). https://doi.org/10.1016/J.COMPAG.2019.05.049 1, 3
29. Đukic, N., Lukezic, A., Zavrtanik, V., Kristan, M.: A low-shot object counting network with iterative prototype adaptation. In: ICCV. pp. 18826–18835. IEEE (2023). https://doi.org/10.1109/ICCV51070.2023.01730 3, 7, 8
30. Wang, X., Girdhar, R., Yu, S.X., Misra, I.: Cut and learn for unsupervised object detection and instance segmentation. In: CVPR. pp. 3124–3134 (2023) 7
31. Wang, Z., Xiao, L., Cao, Z., Lu, H.: Vision transformer off-the-shelf: A surprising baseline for few-shot class-agnostic counting. In: AAAI. pp. 5832–5840. AAAI Press (2024). https://doi.org/10.1609/AAAI.V38I6.28396 3, 7, 8

32. Xue, Y., Ray, N., Hugh, J., Bigras, G.: Cell counting by regression using convolutional neural network. In: ECCV. vol. 9913, pp. 274–290. Springer (2016). https://doi.org/10.1007/978-3-319-46604-0_20 1, 3
33. Zhang, S., Wu, G., Costeira, J.P., Moura, J.M.F.: Fcn-rlstm: Deep spatio-temporal neural networks for vehicle counting in city cameras. In: ICCV. pp. 3687–3696. IEEE Computer Society (2017). https://doi.org/10.1109/ICCV.2017.396 1, 2