OCCAM: Class-Agnostic, Training-Free, Prior-Free and Multi-Class Object Counting

Michail Spanakis[1,2]
mikespan@ics.forth.gr
Iason Oikonomidis[1]
oikonom@ics.forth.gr
Antonis Argyros[1,2]
argyros@ics.forth.gr

[1] Institute of Computer Science (ICS)
Foundation for Research & Technology – Hellas (FORTH)
Heraklion, Greece

[2] Computer Science Department
University of Crete
Heraklion, Greece

Abstract

Class-Agnostic object Counting (CAC) involves counting instances of objects from arbitrary classes within an image. Due to its practical importance, CAC has received increasing attention in recent years. Most existing methods assume a single object class per image, rely on extensive training of large deep learning models, and address the problem by incorporating additional information, such as visual exemplars or text prompts. In this paper, we present OCCAM, the first training-free approach to CAC that operates without the need for any supplementary information and addresses the multi-class variant of the problem. Our approach is capable of counting the object instances in each and every class among arbitrary object classes within an image. We leverage Segment Anything Model 2 (SAM2), a foundation model, and a custom threshold-based variant of the First Integer Neighbor Clustering Hierarchy (FINCH) algorithm to achieve competitive performance on widely used benchmark datasets, including FSC-147 and CARPK, while making fewer assumptions. We also propose a synthetic multi-class dataset and the F1 score as a more suitable evaluation metric. The code for our method and the proposed synthetic dataset are available at: https://mikespanak.github.io/OCCAM_counter.

1 Introduction

Object counting is a natural and intuitive ability. Counting fundamentals emerge in humans from a very young age - often even before the development of speech [41]. But what happens when the number of objects becomes large? Counting becomes tedious and repetitive. Despite its apparent simplicity, counting remains a cognitively demanding task. It constitutes a step after object recognition, localization, and differentiation, making it especially challenging for computers.

Despite these challenges, numerous approaches have been developed for counting objects in images. Traditional computer vision techniques were used in early methods, *e.g.* [17]. In recent years, class-specific object counting methods have achieved significant progress utilizing prior knowledge regarding the target object class due to advances in neural networks and deep learning. Specialized models have been developed for counting humans in

© 2026. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

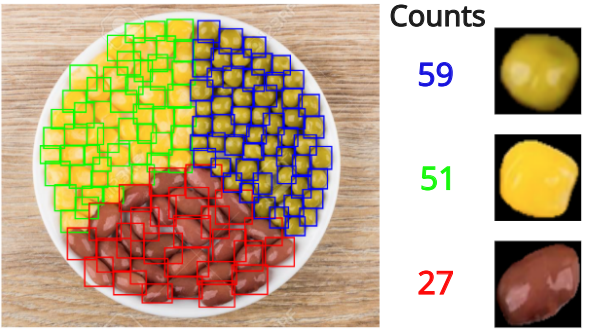

Figure 1: We propose the first Training-free, Prior-free and Multi-class CAC approach that takes as input an RGB image and outputs a count of object instances per detected class.

crowded scenes [6], vehicles [11], animals [3], and other categories of objects in various domains [31, 45, 47]. Marsden *et al*. [23] proposed a CNN-based counter that can be adapted to different visual domains and object types. These counting models can reliably count a large number of objects belonging to a single predefined class. As long as sufficiently large annotated training datasets are available for each object class, there is no inherent restriction on the target object class. However, collecting and annotating these datasets is often prohibitively expensive. Furthermore, most existing approaches require training a separate model for each category of objects, which limits their scalability and flexibility. These limitations have led to the emergence of a more generalized and demanding research direction: class-agnostic object counting [4, 21].

To address the aforementioned limitations, approaches have been developed to count objects from arbitrary object classes. These are typically trained on images including several objects from various categories to learn the underlying notion of similarity between object instances. However, most existing methods assume the presence of only a single class per image. As a result, they rely on additional input, such as visual exemplars or text prompts, to specify which objects should be counted. Some studies have introduced interactive counting frameworks [2, 13] that incorporate human feedback into the loop to refine the predictions. Other approaches [32] count only the object class with the highest number of instances. Although these methods represent important progress in the field, they still face challenges in terms of generalization and practical applicability to diverse real-world scenarios.

Recent approaches omit the dependence on extensive training by leveraging pre-trained, general-purpose vision models, such as SAM [16] and DINOv2 [26]. In addition, other methods [9, 32] operate without requiring any form of prior information. Some methods revoke the assumption of a single object class per image [15, 29, 43] or aim to count instances from multiple classes [10, 24]. Our method, **O**bject **C**ounting: **C**lass-agnostic, **A**ssumption-minimal, **M**ulti-class (**OCCAM**), advances this research direction by imposing, to the best of our knowledge, the fewest assumptions to date. Specifically, our method is: (1) **Training-free**: it does not depend on some kind of training procedure, allowing immediate out-of-the-box deployment; (2) **Prior-free**: only requires a plain RGB image, since additional information implicitly defines the target object category a priori; (3) **Multi-class**: it is capable of counting multiple object categories within an image in a single pass.

Solving multi-class CAC without restrictive assumptions or prior knowledge is crucial because it addresses the fundamental limitations of scalability, generalization, and practical

deployment in real-world scenarios. Traditional approaches rely on priors, such as exemplars and text prompts, or the assumption of a single dominant class, to define what should be counted. This constrains flexibility, introduces bias, and creates bottlenecks for fully automated systems. These requirements are particularly limiting in complex environments where object categories are unknown and diverse. By removing both prior information and task-specific training, a method can operate in a truly open-world setting, autonomously discovering and counting arbitrary object categories directly from raw images. This enables out-of-the-box performance, supports simultaneous multi-class counting, and makes such approaches far more suitable for scalable, general-purpose, and fully automated applications.

To achieve the above, OCCAM leverages SAM2 [34] and our proposed threshold-based variant of the First Integer Neighbor Clustering Hierarchy (FINCH) algorithm [35]. Extensive experimental evaluation demonstrates the effectiveness and generalizability of our approach in several datasets. To ease the evaluation of the multi-class aspect of OCCAM, we introduce a new synthetic dataset. Furthermore, we discuss the evaluation metrics adopted in the relevant literature. We argue that the F1 score is more suitable for evaluating CAC methods, since it does not take into account only the numbers of the identified objects but also their corresponding correctness. OCCAM achieves competitive and, in some cases, state-of-the-art performance across standard metrics. Currently, there are no other CAC methods that are both prior-free and training-free. All these properties combined make OCCAM truly class-agnostic, ideal for fully automated object counting systems, and position it as the most general solution available.

In summary, the key contributions of our work are the following:

- We introduce the first CAC approach that is simultaneously training-free, prior-free, and capable of counting object instances from multiple classes within an image.
- We present a synthetic multi-class dataset for object counting, constructed by stitching together an arbitrary number of real-world single-class images.
- We propose the F1 score as a more suitable and unbiased evaluation metric for CAC methods, enhancing the validity of the quantitative evaluation.

2 Related Work

Class-Agnostic object Counting (CAC) generalizes object counting across arbitrary categories, reducing the reliance on class-specific training. Existing CAC methods can be grouped by (i) whether they depend on priors (*e.g.* exemplars, points, or text prompts), (ii) whether they require dedicated training, and (iii) whether they assume a single dominant class per image or support multi-class counting.

Prior-Based CAC. Most CAC methods use priors to specify what to count during training and inference. Exemplar-based methods leverage the strongest guidance via 1-3 cropped instances from the given image. These provide crucial information regarding the depicted objects, such as their shape, size, and color, or, even more subtle, yet important information, such as perspective and lighting condition. Counting is typically cast as matching or similarity learning, from the first self-similarity formulation in [21] proposing GMN, to few-shot regression-based FamNet [33], and BMNet/BMNet+ [36] which jointly learns feature representation and a similarity metric. Nguyen *et al.* [25] proposed Counting-DETR,

an uncertainty-aware detector trained via pseudo labels, while SAFECount [44] refines separation of objects through similarity-aware comparison and feature fusion. Text-prompted CAC improves generality by weakening the prior, enabling counting from class labels [42]. Recently, multi-modal methods have been developed, often building on foundation models. CounTR [19] is a transformer-based model that leverages attention to compare image patches with exemplars. LOCA [7] uses an iterative Prototype Extraction module to fuse the exemplar shape and appearance with the image features. PseCo [14] combines SAM [16] and CLIP [30] in a point–segment–count pipeline, while MAFEA [15] mitigates target confusion by cross-attention. In DAVE [29] density-based candidate generation is combined with a detection-based verification stage to improve localization and outlier removal. A single-stage counter GeCo [28] unifies detection, segmentation, and counting. COUNTGD [1] utilizes GroundingDINO [20] fusing text and visual prompts via attention.

Prior-Free CAC. A smaller body of work [9, 32, 40, 46] avoids external priors and instead exploits self-similarity and internal feature relationships. RepRPN-Counter [32], the first exemplar-free CAC method, discovers repeating objects via an RPN and estimates counts through density prediction. RCC [9] frames counting as repetition recognition under weak supervision. GCA-SUN [40] directly predicts density maps using gated context-aware modulation, feature selection, and multi-scale fusion, while SAVE [46] uses detection-based features, self-attention over visual embeddings, and count regression. Although prior-free approaches are suitable for automation, the ambiguity of what to count becomes more prominent, making the single-class-per-image assumption essential. Furthermore, some multi-modal methods [7, 15, 19, 28, 29] can operate without additional information, typically trading off counting accuracy for generality.

Training-Free CAC. Most CAC methods still require extensive task-specific training of advanced deep learning models and large-scale datasets, making them resource-intensive. Training-free methods [12, 18, 22, 27, 37, 39] aim to overcome these limitations by leveraging foundation models. Ma *et al*. [22] are the first to use SAM [16] exploiting exemplar-guided mask generation and cosine-similarity-based feature averaging for counting. TFOC [37] also uses SAM, formulating counting as prior-guided segmentation integrating similarity, segment, and semantic priors. TFCounter [39] utilizes a dual-point prompt strategy and a context-aware similarity module, while in [18] instance-level counting with improved point prompting and semantically enriched features is performed by SAM. CountingDINO [27] relies on fully unsupervised DINOv2 [26] features to extract latent object prototypes and generate similarity-based density maps. Count validity is improved by combining SAM-based spatial proposals with EVA-CLIP–based [38] semantic filtering in [12] ensuring that only spatially and semantically relevant object instances are counted.

Multi-Class CAC. Real-world images typically contain multiple categories of objects. Some methods [15, 29, 43] have been developed to utilize priors to count object instances of the target class despite the presence of multiple categories. Xu *et al*. [43] combine exemplar-based segmentation with pseudo-labeled masks to mitigate multi-class over-counting and highlight the trade-off between handling multiple classes and single-class counting accuracy.

Only recent approaches are capable of counting object instances belonging to every class within an image. ABC123 [10], the first multi-class CAC method, is a transformer-based density regression approach trained on the synthetic MCAC dataset to simultaneously count multiple object types without priors. Each object class can have up to 300 object instances

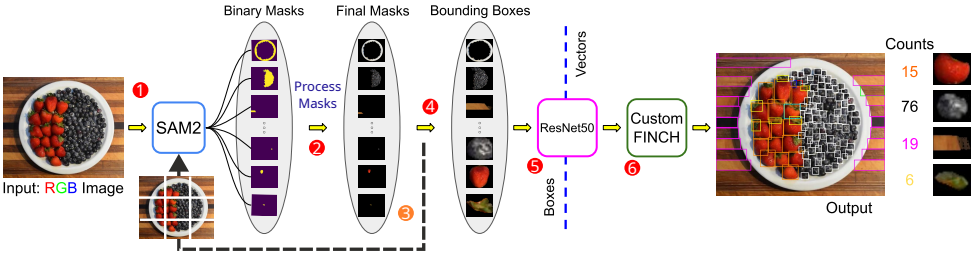

Figure 2: Given an RGB image as input, a dense grid of seed points is constructed and passed to SAM2 to produce a collection of binary masks (**Step 1**). These masks are then filtered to obtain a refined set of candidate object masks (**Step 2**). If very few masks are left, the Multiscale Refinement Strategy is triggered (**Step 3**). Each candidate object mask is used to extract an aspect-ratio-preserving bounding box (**Step 4**), which is passed through ResNet50 to produce a feature vector (**Step 5**). These vectors are then grouped using a custom threshold-based variant of FINCH algorithm (**Step 6**), resulting in a final set of clusters.

and at most 4 different categories can be counted per image. OCCAM overcomes these limitations despite being training-free; it can count objects belonging to an arbitrary number of different classes without an upper bound on their instances. OmniCount [24] is training-free and detection-based, using SAM for instance segmentation, relying on semantic and geometric priors to count multiple categories in a single pass. In contrast, OCCAM does not rely on provided priors. Overall, OCCAM is uniquely positioned in the field as the most general CAC method to date, with no direct competitors. Every comparison should be made taking into consideration the aforementioned differences.

3 Methodology

We aim to address training-free and multi-class Class-Agnostic object Counting (CAC) without using priors. In order to achieve this, our method, Object Counting: Class-agnostic, Assumption-minimal, Multi-class (OCCAM), is based on two key components: the Segment Anything Model 2 (SAM2) [34] and a threshold-based variant of the First Integer Neighbor Clustering Hierarchy (FINCH) algorithm [35] that we propose. Figure 2 illustrates the proposed approach providing an overview of the major processing steps.

Object Mask Generation. We utilize Segment Anything Model 2 (SAM2) [34] to obtain masks for each object instance in the image without prior knowledge of their categories, number, or size (**Step 1**). Specifically, since the RGB image $I \in \mathbb{R}^{H \times W \times 3}$ is our sole form of input, we densely sample a custom canonical grid of possible seed points, p_i, over I and use them as point-prompts for SAM2. Each p_i may correspond to a distinct object of interest. The spacing between the seed points is chosen empirically and remains constant across all images, resulting in a regular grid whose size depends on the dimensions of I. For each p_i, SAM2 produces up to three binary masks, M_i, without requiring any additional information.

Object Mask Postprocessing. Further processing of the raw binary masks M_i is necessary, as there is no guarantee that each mask produced by SAM2 corresponds to a distinct and unique object instance (**Step 2**). There are two main issues: (1) different seed points may

produce the same, or a slightly different, binary mask, and (2) SAM2 may produce one to three different masks from the same seed point. Therefore, each pixel may not be uniquely assigned to a single mask, which can adversely affect the final object count.

In order to resolve these issues, we proceed to mask filtering. Regarding the first issue, duplicate masks are discarded. Specifically, masks are considered "duplicates" if their Intersection over Union (IoU) value exceeds an empirically determined threshold, in which case only one mask is retained arbitrarily. Enforcing SAM2 to produce a single mask per seed point would not resolve the second issue, as the target class is not specified a priori due to the lack of additional information. We rely on our proposed FINCH variant to overcome this problem. However, we try to mitigate it by applying more filtering rules. If a mask consists of multiple connected components, only the major one is kept, unless it constitutes a duplicate with a different mask produced independently. This is crucial because the produced masks often contain a couple of distinct objects, each of which could be a separate target object. As a result, only a single object instance is contained in each mask, any occlusive object is excluded, and the size of each segmented object can be estimated more accurately.

The estimated size of each object instance is utilized in the following filtering rules. Final binary masks that are too big, relative to the image size, are discarded, since they may correspond to the background or be a super-set of other masks. Additionally, masks with a single-pixel width or height are discarded, as they are likely too thin to represent valid objects and may be artifacts. In general, these filtering rules eliminate some binary masks, M_i, resulting in the final set of candidate object masks, $\hat{M}_j$, which, ideally, correspond to distinct objects of interest.

Multiscale Refinement Strategy. If the final set of candidate objects contains very few masks, a scaling scheme similar to a test-time normalization technique used in [19] is utilized to enable SAM2 to detect small objects (**Step 3**). The given image, I, is split into a non-overlapping canonical 3×3 grid, and the entire methodology is recursively applied to each of the produced sub-images, independently. Filtered binary masks produced from each of the nine sub-images are accumulated to form the final set of candidate objects, $\hat{M}_j$. This strategy enables the model to detect smaller objects that may be overlooked in the original image. If no additional objects are detected using this scaling scheme, we conclude that this image contains very few objects. Through this mechanism, OCCAM can differentiate between uniform images, or empty scenes, and images containing numerous tiny objects.

Feature Extraction. The next step is to group the final set of candidate object masks, $\hat{M}_j$. First, each retained mask, $\hat{M}_j$, is applied to the original image to extract the corresponding colored region, and tight bounding boxes are formed (**Step 4**). In order to avoid any size-related bias, possibly caused due to perspective, each bounding box is resized to a predetermined size, preserving the original aspect ratio of the depicted object. Any residual space of each bounding box is zero-padded. Consequently, each of these aspect-ratio-preserving bounding boxes is color-normalized and passed through a deep model for a standardized representation to be extracted (**Step 5**). To achieve this, the standard ResNet-50 [8] pre-trained on ImageNet [6] excluding its last fully connected layer is utilized. Therefore, this procedure results in a 2048-dimensional feature vector, $f_j \in \mathbb{R}^{2048}$, produced for each bounding box.

Threshold-based FINCH Variant. A custom threshold-based variant of the First Integer Neighbor Clustering Hierarchy (FINCH) algorithm [35] is proposed to properly group the

extracted feature vectors, f_j, into an arbitrary number of clusters (**Step 6**). Each of these clusters should ideally correspond to a distinct object category in the image.

The key idea of FINCH is to iteratively group vectors which are the nearest neighbors to each other, based on a chosen distance metric. However, the nearest-neighbor relation is generally asymmetric; a vector v_i may be the nearest neighbor of another vector v_j, but this does not necessarily imply that v_j is also the nearest neighbor of v_i. This may occur as v_i may have a nearer neighbor in a different direction. FINCH bypasses this issue by enforcing symmetry, effectively allowing multiple "nearest" neighbors. Hence, in our example, v_i is also marked as the nearest neighbor of v_j, resulting in a situation where v_j has two nearest neighbors. Furthermore, the absolute value of the distance is not taken into consideration, allowing the algorithm to remain completely parameter-free. Finally, the minimum possible size of a cluster is two, which could cause various issues in our case, such as the adverse grouping of the feature vector corresponding to the background of the image.

In our approach, the standard FINCH algorithm is modified by incorporating an empirical threshold in each of its steps. Thus, two vectors will not be grouped together if the distance between them is greater than this threshold, despite their partial nearest-neighbor relation. In all of our experiments, the Euclidean distance is used in the 2048-dimensional feature space. For the first iterations, the corresponding threshold decreases in each step, while it remains constant for the later iterations, balancing intra-class variance and inter-class discrimination. This is a simple but effective way to prevent the enforcement of symmetry and resolve the issues mentioned above.

Initially, each vector forms a cluster by itself. After the first iteration, the centroids of the formed clusters are estimated and the distances among the clusters are computed based on these centroids. Two clusters are merged if they are, at least, partial nearest neighbors and their centroids are at a distance less than the defined threshold. After any merging that occurs, the centroids of the newly formed clusters are estimated. The iterative procedure is terminated when there are no more clusters left to be merged. Compared to the original FINCH algorithm, this constitutes a different termination condition, since FINCH stops producing hierarchical clustering partitions when the next partition would include every given element. By utilizing a proper threshold, we eliminate the need to devise a non-trivial automatic procedure for selecting the optimal clustering level from the hierarchical partitions generated by the standard FINCH algorithm.

The output of our proposed threshold-based FINCH variant is the resulting clusters, C_k, each of which corresponds to a number of candidate object masks in the input image. The cardinality of each cluster constitutes the final count of objects. Since our approach is prior-free and multi-class, the paradigm introduced in [10] is followed in order for the user to be informed of the results. Specifically, for each final cluster, the representative image is the bounding box that corresponds to the closest vector to its estimated centroid.

4 Experiments

Datasets. *FSC-147 Dataset:* The FSC-147 dataset [33] is the benchmark for few-shot, class-agnostic object counting, consisting of 6135 human-annotated images from 147 object classes with non-overlapping object categories in train/val/test splits. The test set contains 1190 images from 29 diverse classes, from sea shells to stamps, with per-instance center annotations and three exemplar bounding boxes per image. The average count is 56 objects per image. Since our method is training-free and prior-free, only the test set is used, without

Model	Info	MAE ↓	RMSE ↓	NAE ↓	SRE ↓
Mean	-	47.55	147.67	-	-
Median	-	47.73	152.46	-	-
OmniCount [24]	Exemplars	18.63	112.98	**0.14**	2.99
OmniCount [24]	Points	19.24	115.27	0.25	3.21
OmniCount [24]	Text	21.46	133.28	0.32	**0.39**
OCCAM-S	-	**16.92**	**110.83**	0.22	3.38
ABC123 [10]	-	11.75	33.41	**0.11**	**2.38**
OCCAM-S ($\leq$ 300)	-	**11.29**	**25.17**	0.21	2.68

Table 1: Multi-class Methods on the FSC-147 Test Set (best results in **bold**). Results reported below the double line include images with $\leq$ 300 objects.

relying on exemplars or annotations. A key limitation of FSC-147 is the imbalanced class distribution and the single-class-per-image assumption, which is not realistic and does not hold for many images. The common heuristic that the most frequent class is the annotated one is not always valid. Thus, this dataset is less suitable for evaluating multi-class methods.

Synthetic Multi-class Test Set: We construct a synthetic multi-class test set by stitching together randomly selected images from the FSC-147 test set. Three dataset variants are generated to assess robustness and compensate for the introduced randomness, each containing 100 images composed of 1–10 sub-images depicting a different and distinct object class. The distribution of these classes is also random but not uniform, inheriting the aforementioned distribution bias of the FSC-147 test set. Sub-images are stitched side by side using the normalized height of FSC-147, forming grids of at most two columns without resizing, resulting in quite large synthetic images. Unused regions are zero-padded. The class composition and instance counts are unconstrained, resulting in images with high object densities. All of the above make the proposed dataset challenging and suitable for multi-class counting.

Real Multi-class Test Set: A recently proposed dataset [43] has been acquired by privately contacting the respected authors. It has been created in order to examine the recently discovered tendency of many CAC methods to count every object instance present in an image [15, 29, 43], regardless of its class. It contains 453 real-world images, each depicting 1–3 object classes from 76 different object categories with possible inter-class overlap. Three exemplars for each object class and point-annotations per object instance are provided in order to be in accordance with FSC-147 and easily usable by earlier approaches. Its major advantage is that object instances are annotated across all classes present in each image, making this dataset well suited for evaluating multi-class CAC methods.

CARPK Dataset: CARPK [11] is a counting benchmark consisting of 1448 aerial (top-down) drone images. Cars, the only annotated class, are annotated by tight bounding boxes. The test set consists of 459 images. In the CAC field, it is utilized to examine the ability of the proposed approaches to generalize adequately to domain-specific counting tasks.

Implementation Details. We identify two slightly different configurations of our methodology, one tailored for the single-class case (**OCCAM-S**) and one for the multi-class case

Model	Info	MAE ↓	RMSE ↓
CAN SAM? [22]	3-shot	27.97	131.24
CountingDINO [22]	3-shot	20.93	71.37
TFOC [37]	3-shot	19.95	132.16
ValidCounter [12]	3-shot	19.33	133.33
TFCounter [39]	3-shot	18.56	130.59
A-Simple-But [18]	3-shot	**12.26**	**56.33**
CountingDINO [22]	1-shot	37.73	131.71
CAN SAM? [22]	1-shot	33.53	142.28
A-Simple-But [18]	1-shot	**15.15**	**69.16**
TFOC [37]	Text	24.79	137.15
A-Simple-But [18]	Text	**23.59**	**113.60**
OCCAM-S	-	**16.92**	**110.83**

Table 2: Training-free Methods on the FSC-147 Test Set (best results in **bold**).

Model	MAE ↓	RMSE ↓
RepRPN-C [32]	26.66	129.11
RCC [9]	17.12	104.53
LOCA [7]	16.22	103.96
DAVE [29]	15.14	103.49
CounTR [15]	14.71	106.87
GCA-SUN [40]	14.00	92.19
GeCo [28]	13.30	108.72
MAFEA [15]	13.23	105.99
SAVE [46]	**8.92**	**80.39**
OCCAM-S	16.92	110.83

Table 3: Prior-Free Methods on the FSC-147 Test Set (best results in **bold**). Scores for RepRPN-Counter [32] are reported for "Top 5".

(**OCCAM-M**). Both rely on the "SAM 2.1 Large" backbone, introduced in [34]. Moreover, a seed point is set per 10 pixels and an IoU threshold of 0.1 is used for both. The differences of the two configurations are: (1) the size of the bounding boxes is set to 224×224 in OCCAM-S and to 500×500 in OCCAM-M, (2) the different thresholding strategies in our proposed variation of the FINCH algorithm. OCCAM-M uses a lower threshold in each iteration of FINCH, in order to be more strict and differentiate between objects that belong to different categories. On the other hand, if numerous objects of the same class are depicted in the given image, our approach should be less strict. Toward this end, OCCAM-S uses a higher threshold in each iteration in order to group objects despite some minor differences in their appearance allowing more intra-class variance. This trade-off between inter-class discrimination and tolerance to intra-class appearance variation is also highlighted in [23].

Evaluation Metrics. Our method is evaluated using the most widely adopted metrics in the CAC literature: Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE),

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{k_i}\sum_{j=1}^{k_i}|y_{ij}^{gt} - y_{ij}| \quad \text{and} \quad RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\frac{1}{k_i}\sum_{j=1}^{k_i}(y_{ij}^{gt} - y_{ij})^2}, \tag{1}$$

as introduced in [21, 33]. In the above definitions, n is the number of test images in the dataset and k_i is the number of classes in the i^{th} image. The ground-truth and the predicted number of object instances of the j^{th} object class in the i^{th} image are y_{ij}^{gt} and y_{ij}, respectively.

However, the significance of an absolute counting error depends heavily on the number and appearance of the depicted ground-truth objects. For instance, a MAE of 10 represents a greater relative error in an image containing only 10 distinct large objects than in one containing 1000 small and heavily occluded objects. To account for this, Normalized Absolute Error (NAE) and Squared Relative Error (SRE),

$$NAE = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{k_i}\sum_{j=1}^{k_i}\frac{|y_{ij}^{gt} - y_{ij}|}{y_{ij}^{gt}} \quad \text{and} \quad SRE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\frac{1}{k_i}\sum_{j=1}^{k_i}\frac{(y_{ij}^{gt} - y_{ij})^2}{y_{ij}^{gt}}}, \tag{2}$$

Model	MAE ↓	RMSE ↓	NAE ↓	SRE ↓	Precision ↑	Recall ↑	F-1 ↑
FamNet+ [33]	22.08	99.54	0.44	6.45	0.66	0.67	0.57
BMNet+ [36]	14.62	91.83	0.27	**6.19**	0.66	0.68	0.67
CounTR [19]	11.95	91.23	**0.23**	7.44	**0.79**	**0.78**	**0.78**
LOCA [2]	**10.79**	**56.97**	0.26	7.96	0.78	**0.78**	**0.78**
TFOC [37]	19.95	132.16	0.29	3.80	0.79	**0.82**	0.80
ValidCounter [12]	19.33	133.33	**0.20**	3.43	0.87	0.77	0.82
OCCAM-S	**16.92**	**110.83**	0.22	**3.38**	**0.89**	**0.82**	**0.83**

Table 4: Precision, Recall and F-1 Score on the FSC-147 Test Set (best results in **bold**). Our approach is Prior-free, while all other methods use exemplars. Below the double line, the approaches are Training-free. NAE, SRE, Precision, Recall and F-1 scores for the other methods are provided in [12].

which normalize the error relative to the number of objects in each image, are also reported. Since our approach is multi-class, the adjusted versions, introduced in [10], are employed.

The above metrics can be misleading. Each ground-truth object is represented by a point annotation at its approximate center, and a predicted mask is considered a True Positive if it contains exactly one point annotation. Thus, a perfect numerical score does not ensure that all counted instances correspond to ground-truth objects, as False Positives (masks containing no ground-truth annotations or multiple annotations) and False Negatives (ground-truth annotations that are not contained in any properly clustered mask) may coincidentally cancel out. To mitigate this, we focus on True Positives and adopt Precision, Recall, and the F1 Score, which is unbiased. These metrics correspond to their standard definitions and are well-established in related domains. They provide a more reliable and interpretable assessment of counting performance, ensuring that the predicted instances correspond to valid objects, especially for images with objects from multiple categories. They are also used in the foundational paper [21] and their relevance is further emphasized in [12], a recent work concurrent with ours.

Quantitative Results. Despite not having a direct competitor due to its parsimonious assumptions, OCCAM is competitive with other multi-class and training-free approaches. However, our approach is lacking behind the prior-free ones, each of which relies on training. Indicative of its success are the state-of-the-art Precision, Recall, and F1 scores on the FSC-147 Test Set. According to the obtained results, OCCAM has the lowest MAE for the Real Multi-class Test Set and generalizes well on the CARPK dataset. Overall, OCCAM performs remarkably well, given its minimal assumptions.

Regarding the quantitative evaluation on the FSC-147 Test Set, our approach is compared with the mean and median baselines and the few other multi-class methods in table 1. Our approach outperforms the baselines by a significant margin. OmniCount [24] is our main competitor, despite requiring some form of additional information, since it is the only multi-class method that is training-free, detection-based, and evaluated on the whole FSC-147 Test Set, similarly to our method. In this comparison, OCCAM-S achieves better MAE and RMSE scores without utilizing any priors, while the NAE and SRE scores are also competitive. The results of ABC123 [10], which is trained on the MCAC dataset, are reported on the FSC-147 Test Set without any further tuning and only for images with at most 300 ob-

Figure 3: Qualitative results from the FSC-147, the Real Multi-class, and our proposed Synthetic Multi-class test sets. The fourth line of results presents some failure cases.

jects. OCCAM-S achieves better MAE and RMSE scores than ABC123 and has competitive NAE and SRE, despite being training-free. In addition, the impact of a few images of the FSC-147 Test Set which contain more than 300 objects becomes evident by their exclusion. In this case, MAE and RMSE are significantly reduced.

OCCAM-S is compared with other training-free methods on the FSC-147 Test Set in table 2. Every other method is single-class and utilizes some sort of prior (3 exemplars, 1 exemplar, or text prompts). However, our approach, despite being prior-free, surpasses all methods except A-Simple-But [18] in terms of the MAE score, and achieves a competitive RMSE score. Our approach is compared with other prior-free methods on the FSC-147 Test Set, in table 3. It is emphasized that our approach is the only multi-class and training-free method among the compared ones. In table 4, OCCAM-S is compared with other methods regarding Precision, Recall, and F1-Score on the FSC-147 Test Set. All other methods are single-class and utilize exemplars as additional information. TFOC [37] and ValidCounter [12] are also training-free. Despite making the least assumptions, our approach achieves competitive scores, having state-of-the-art Precision, Recall, and F1-Score.

Training Set	FSC-147		FSC-147 & Synthetic	
Model	MAE ↓	RMSE ↓	MAE ↓	RMSE ↓
FamNet [33]	13.03	20.28	11.03	17.60
FamNet+ [33]	19.42	39.78	11.31	18.84
BMNet+ [36]	25.55	40.35	11.44	23.22
SAFECount [44]	23.57	40.99	9.80	32.40
Seg-Count [43]	-	-	6.97	**13.03**
OCCAM-S	16.82	42.21	16.82	42.21
OCCAM-M	**5.52**	**13.43**	**5.52**	13.43

Table 5: Methods on the Real Multi-Class Test Set (best results in **bold**). The left pair of columns corresponds to methods trained only on FSC-147, while the right pair to methods trained on FSC-147 and the synthetic dataset introduced in [43]. Our approach is Training-free, Prior-free and Multi-class. The other methods have none of these properties.

We evaluate our approach on several additional datasets. In table 5, our approach is compared with other methods in the Real Multi-Class Test Set [43]. Our approach is Training-free, Prior-free and Multi-class, while the rest of the compared methods have none of these properties. OCCAM-S achieves competitive MAE and RMSE scores when the other approaches are trained solely using the FSC-147 training set. OCCAM-M achieves state-of-the-art MAE and the second best RMSE. The NAE, SRE, Precision, Recall, and F1-score of OCCAM-M are 0.24, 1.85, 0.90, 0.81, and 0.83, respectively (not shown in the table). OCCAM is compared with other CAC methods on the CARPK dataset in table 6. Our method achieves competitive scores across all metrics. The OCCAM-S configuration has state-of-the-art Precision, 0.90, Recall, 0.89, and F-1 Score, 0.89. The second best method being the ValidCounter [12] scoring 0.87, 0.84, and 0.86, respectively (not shown in the table). In table 7, our approach is evaluated on our proposed Synthetic Multi-class Test Set. Over the 3 different variants of the dataset, the robustness of our approach is examined and the mean performance is computed for each configuration, OCCAM-S and OCCAM-M. There is no significant difference in performance between each dataset variant. On this multi-class dataset, OCCAM-M achieves better results overall, as expected.

Qualitative Results. We present indicative qualitative results in fig. 3. Tight bounding boxes are drawn on the counted objects in the original RGB image, with different colors denoting different clusters. Therefore, the exact location of each counted object instance is provided, and the produced results can be assessed visually, aiding interpretability.

In the top left image, our approach counts correctly the strawberries in this fairly complex scene. In the middle image of the top row, the found clusters are separated properly, while in the top right image, our model not only counts what is on the plate, but also counts the green circles painted on the plate and the 4 marks of the wooden surface. Given that the images in the top row are from the FSC-147 Test Set, we observe that the single-class-per-image assumption does not hold. In the first image of the second row, our approach discriminates and counts the apples and the leaves, despite apples being very diverse in color and size. OCCAM, in the middle image of this row, counts both the entire lego blocks and the protrusions of each block. The right image of the second row is a rather simple image from

Model	MAE ↓	RMSE ↓	NAE ↓	SRE ↓
TFOC [37]	10.97	14.24	0.48	3.70
OmniCount [24]	9.92	12.15	0.23	2.11
TFCounter [39]	9.71	12.44	-	-
ValidCounter [12]	9.36	12.86	**0.17**	**1.41**
A-Simple-But [18]	**4.39**	**5.70**	-	-
GCA-SUN [40]	10.96	13.95	-	-
SAVE [46]	9.56	19.04	-	-
RCC [9]	**9.21**	**11.33**	-	-
OCCAM-M	14.83	21.96	0.21	2.20
OCCAM-S	**10.06**	**13.81**	**0.13**	**1.41**

Table 6: Results on the CARPK Test Set (best in **bold**). RCC [9] is fine-tuned on CARPK. Methods above the double line are Training-free, but utilize exemplars, while below the line are Prior-free, but rely on training. Only our approach and Omnicount [24] are Multi-class.

Variant	MAE ↓	RMSE ↓	NAE ↓	SRE ↓	Precision ↑	Recall ↑	F1 ↑
1	29.20	54.51	0.82	9.03	0.75	0.72	0.68
2	26.56	53.89	0.87	10.16	0.78	0.76	0.72
3	31.89	62.91	0.97	11.33	0.74	0.72	0.67
Mean	29.22	57.11	0.89	10.18	0.75	0.73	0.69
1	21.99	37.45	0.42	4.17	0.85	0.59	0.66
2	19.81	34.33	0.38	3.60	0.88	0.62	0.69
3	22.28	39.16	0.44	4.18	0.85	0.59	0.66
Mean	21.36	36.98	0.41	3.99	0.86	0.60	0.67

Table 7: Performance of OCCAM-S (above the double line) and OCCAM-M (below) on our proposed Synthetic Multi-class Test Set.

our proposed Synthetic Multi-Class Test Set consisting of 3 categories, which are counted by our approach. In the left image of the third row, the discriminative ability of our method is on display as the strawberries are grouped in two clusters based on where their cut side faces. In the middle image, our approach even counts the watermarks. In the right image of the third row, OCCAM correctly discriminates and counts the three depicted classes.

Finally, the fourth row presents some failure cases. Incorrectly splitting or merging clusters are typical issues affecting the final object count. This issue is inherently ambiguous since, in many cases, the desirable result is not well defined: without any prior information specifying the desired semantic grouping, multiple valid interpretations of object similarity may exist. This issue is not eliminated even under the oversimplifying single-class-per-image assumption. In the left image, a coherent cluster, the stamps, is split, but there is some insight. The stamps are divided into horizontal (light blue boxes) and vertical (fuchsia boxes). In the middle image, there is a cluster of sunglasses and a cluster of individual lenses. However, given that our approach is prior-free, the exact objects of interest cannot be specified. Thus, visually meaningful but not so useful clusters are sometimes formed. In

M	C	S	MAE ↓	RMSE ↓	NAE ↓	SRE ↓	Precision ↑	Recall ↑	F-1 ↑
✗	✗	✗	78.83	153.86	2.35	14.24	0.36	0.92	0.49
✗	✗	✓	78.74	153.74	2.35	14.23	0.36	0.92	0.49
✗	✓	✗	46.38	141.88	0.99	8.02	0.51	0.81	0.59
✗	✓	✓	46.19	141.69	0.99	8.01	0.51	0.81	0.59
✓	✗	✗	28.27	134.99	0.68	5.58	0.66	0.90	0.73
✓	✗	✓	25.38	111.10	0.68	5.29	0.66	0.90	0.74
✓	✓	✗	19.83	134.76	**0.22**	3.80	**0.89**	0.81	**0.83**
✓	✓	✓	**16.92**	**110.83**	**0.22**	**3.38**	**0.89**	**0.82**	**0.83**

Table 8: Ablation Study on the major components of OCCAM-S configuration on the FSC-147 Test Set (best results in **bold**). Columns **M**, **C**, **S** denote **Mask Processing**, **Clustering**, and **Scaling Strategy**, respectively.

the last image, it is not clear whether the target object class is the apples or the bricks on the wall. Our method cannot distinguish every apple due to the perspective of the scene.

Ablation Study. The OCCAM-S configuration and the FSC-147 Test Set are used for the ablation study on the major components of our approach (table 8). According to the scores reported, each component is useful, as removing any of them affects the final performance. Mask Processing is the most impactful component: erroneous and redundant masks are eliminated, reducing the candidate object masks by a large margin and limiting the possible error. As a result, our custom FINCH algorithm has to properly group fewer masks. Clustering further improves the results, since assuming that all objects belong to a single class is unrealistic. The Scaling Strategy contributes specifically in images containing tiny objects. However, these few images have a great impact on MAE and RMSE scores, indicating the unsuitability of these metrics. The above observations are expected for a single-class dataset with densely populated images, given the stage of each component in the pipeline.

5 Conclusions

In this paper, we introduced OCCAM, a novel approach for Class-Agnostic object Counting (CAC). It operates out of the box and addresses key limitations of existing CAC methods by eliminating the need for extensive training and supplementary information, while also supporting the simultaneous counting of multiple object classes. Its minimal assumptions make OCCAM uniquely positioned. By leveraging Segment Anything Model 2 (SAM2) alongside our adapted version of the FINCH algorithm, our approach achieves competitive performance on standard benchmark datasets, such as FSC-147 and CARPK. Additionally, we proposed a synthetic multi-class dataset and emphasized the importance of adopting the F1 score as a more reliable evaluation metric in the field. OCCAM is built as a modular pipeline, allowing each component to be independently replaced as improved models are developed. On this basis, OCCAM constitutes a CAC baseline using minimal assumptions that can serve as a foundation for future research. In future work, a key challenge is to develop a dynamic strategy for estimating the threshold parameters used in FINCH on a per-image basis. We hope that our method serves as a step towards general-purpose and fully automated object counting systems applicable to real-world scenarios.

6 Acknowledgements

This work has been supported by the European Union (EU HE - Magician, Grant Agreement 101120731). The authors also acknowledge support from the VMware University Research Fund (VMURF). In addition, we gratefully acknowledge the assistance of Prof. Dimitris Samaras and his group for providing early access to their Real Multi-class Test Set.

References

[1] Niki Amini-Naieni, Tengda Han, and Andrew Zisserman. Countgd: Multi-modal open-world counting. *NeurIPS*, 37:48810–48837, 2024.

[2] Carlos Arteta, Victor Lempitsky, J Alison Noble, and Andrew Zisserman. Interactive object counting. In *ECCV*, pages 504–518. Springer, 2014.

[3] Carlos Arteta, Victor Lempitsky, and Andrew Zisserman. Counting in the wild. In *ECCV*, pages 483–498. Springer, 2016.

[4] Luca Ciampi, Ali Azmoudeh, Elif Ecem Akbaba, Erdi Sarıtaş, Ziya Ata Yazıcı, Hazım Kemal Ekenel, Giuseppe Amato, and Fabrizio Falchi. A survey on class-agnostic counting: Advancements from reference-based to open-world text-guided approaches. *arxiv:2501.19184*, 2025.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009.

[6] Lijia Deng, Qinghua Zhou, Shuihua Wang, Juan Manuel Górriz, and Yudong Zhang. Deep learning in crowd counting: A survey. *CAAI Trans. on Intelligence Technology*, 9(5):1043–1077, 2024.

[7] Nikola Dukić, Alan Lukežič, Vitjan Zavrtanik, and Matej Kristan. A low-shot object counting network with iterative prototype adaptation. In *ICCV*, pages 18872–18881, 2023.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[9] Michael Hobley and Victor Prisacariu. Learning to count anything: Reference-less class-agnostic counting with weak supervision. *arxiv:2205.10203*, 2022.

[10] Michael Hobley and Victor Prisacariu. Abc easy as 123: A blind counter for exemplar-free multi-class class-agnostic counting. In *ECCV*, pages 304–319. Springer, 2024.

[11] Meng-Ru Hsieh, Yen-Liang Lin, and Winston H Hsu. Drone-based object counting by spatially regularized regional proposal network. In *ICCV*, pages 4145–4153, 2017.

[12] Qinghong Huang, Yifan Zhang, Wenbo Zhang, Jianfeng Lin, Binqiang Huang, Jinlu Zhang, and Wenhao Yu. A training-free framework for valid object counting by cascading spatial and semantic understanding of foundation models. *Information Sciences*, 712:122161, 2025.

[13] Yifeng Huang, Viresh Ranjan, and Minh Hoai. Interactive class-agnostic object counting. In *ICCV*, pages 22312–22322, 2023.

[14] Zhizhong Huang, Mingliang Dai, Yi Zhang, Junping Zhang, and Hongming Shan. Point segment and count: A generalized framework for object counting. In *CVPR*, pages 17067–17076, 2024.

[15] Yerim Jeon, Subeen Lee, Jihwan Kim, and Jae-Pil Heo. Mutually-aware feature learning for few-shot object counting. *Pattern Recognition*, 161:111276, 2025.

[16] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, pages 4015–4026, 2023.

[17] Victor Lempitsky and Andrew Zisserman. Learning to count objects in images. *NeurIPS*, 2010.

[18] Yuhao Lin, Haiming Xu, Lingqiao Liu, and Javen Qinfeng Shi. A simple-but-effective baseline for training-free class-agnostic counting. In *WACV*, pages 8155–8164. IEEE, 2025.

[19] Chang Liu, Yujie Zhong, Andrew Zisserman, and Weidi Xie. Countr: Transformer-based generalised visual counting. *arxiv:2208.13721*, 2022.

[20] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *ECCV*, pages 38–55. Springer, 2024.

[21] Erika Lu, Weidi Xie, and Andrew Zisserman. Class-agnostic counting. In *ACCV*, 2019.

[22] Zhiheng Ma, Xiaopeng Hong, and Qinnan Shangguan. Can sam count anything? an empirical study on sam counting. *arXiv:2304.10817*, 2023.

[23] Mark Marsden, Kevin McGuinness, Suzanne Little, Ciara E Keogh, and Noel E O'Connor. People, penguins and petri dishes: Adapting object counting models to new visual domains and object types without forgetting. In *CVPR*, pages 8070–8079, 2018.

[24] Anindya Mondal, Sauradip Nag, Xiatian Zhu, and Anjan Dutta. Omnicount: Multi-label object counting with semantic-geometric priors. In *AAAI*, pages 19537–19545, 2025.

[25] Thanh Nguyen, Chau Pham, Khoi Nguyen, and Minh Hoai. Few-shot object counting and detection. In *ECCV*, pages 348–365. Springer, 2022.

[26] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arxiv:2304.07193*, 2023.

[27] Giacomo Pacini, Lorenzo Bianchi, Luca Ciampi, Nicola Messina, Giuseppe Amato, and Fabrizio Falchi. Countingdino: A training-free pipeline for class-agnostic counting using unsupervised backbones. *arxiv:2504.16570*, 2025.

[28] Jer Pelhan, Alan Lukezic, Vitjan Zavrtanik, and Matej Kristan. A novel unified architecture for low-shot counting by detection and segmentation. *NeurIPS*, 2024.

[29] Jer Pelhan, Vitjan Zavrtanik, Matej Kristan, et al. Dave-a detect-and-verify paradigm for low-shot counting. In *CVPR*, pages 23293–23302, 2024.

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PmLR, 2021.

[31] Maryam Rahnemoonfar and Clay Sheppard. Deep count: fruit counting based on deep simulated learning. *Sensors*, 17(4):905, 2017.

[32] Viresh Ranjan and Minh Hoai Nguyen. Exemplar free class agnostic counting. In *ACCV*, 2022.

[33] Viresh Ranjan, Udbhav Sharma, Thu Nguyen, and Minh Hoai. Learning to count everything. In *CVPR*, pages 3394–3403, 2021.

[34] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arxiv:2408.00714*, 2024.

[35] Saquib Sarfraz, Vivek Sharma, and Rainer Stiefelhagen. Efficient parameter-free clustering using first neighbor relations. In *CVPR*, pages 8934–8943, 2019.

[36] Min Shi, Hao Lu, Chen Feng, Chengxin Liu, and Zhiguo Cao. Represent, compare, and learn: A similarity-aware framework for class-agnostic counting. In *CVPR*, 2022.

[37] Zenglin Shi, Ying Sun, and Mengmi Zhang. Training-free object counting with prompts. In *WACV*, pages 323–331, 2024.

[38] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arxiv:2303.15389*, 2023.

[39] Pan Ting, Jianfeng Lin, Wenhao Yu, Wenlong Zhang, Xiaoying Chen, Jinlu Zhang, and Binqiang Huang. Tfcounter: Polishing gems for training-free object counting. *arXiv:2405.02301*, 2024.

[40] Yuzhe Wu, Yipeng Xu, Tianyu Xu, Jialu Zhang, Jianfeng Ren, and Xudong Jiang. Gcasun: A gated context-aware swin-unet for exemplar-free counting. *arxiv:2409.12249*, 2024.

[41] Karen Wynn. Addition and subtraction by human infants. *Nature*, 358:749–750, 1992.

[42] Jingyi Xu, Hieu Le, Vu Nguyen, Viresh Ranjan, and Dimitris Samaras. Zero-shot object counting. In *CVPR*, pages 15548–15557, 2023.

[43] Jingyi Xu, Hieu Le, and Dimitris Samaras. Learning to count from pseudo-labeled segmentation. In *WACV*, pages 8754–8763. IEEE, 2025.

[44] Zhiyuan You, Kai Yang, Wenhan Luo, Xin Lu, Lei Cui, and Xinyi Le. Few-shot object counting with similarity-aware feature enhancement. In *WACV*, pages 6315–6324, 2023.

[45] Vitjan Zavrtanik, Martin Vodopivec, and Matej Kristan. A segmentation-based approach for polyp counting in the wild. *Eng. Applications of AI*, 88:103399, 2020.

[46] Ahmed Zgaren, Wassim Bouachir, and Nizar Bouguila. Save: Self-attention on visual embedding for zero-shot generic object counting. *Journal of Imaging*, 11(2):52, 2025.

[47] Zixuan Zheng, Yilei Shi, Chunlei Li, Jingliang Hu, Xiao Xiang Zhu, and Lichao Mou. Rethinking cell counting methods: Decoupling counting and localization. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2024.