Learning to Count from Pseudo-Labeled Segmentation

Jingyi Xu
Stony Brook University
jingyixu@cs.stonybrook.edu

Hieu Le
EPFL
hle@cs.stonybrook.edu

Dimitris Samaras
Stony Brook University
samaras@cs.stonybrook.edu

Abstract

Class-agnostic counting (CAC) has numerous potential applications across various domains. The goal is to count objects of an arbitrary category during testing, based on only a few annotated exemplars. However, existing methods often count all objects in the image, including those from different categories than the exemplars. To address this issue, we propose localizing the area containing the objects of interest via an exemplar-based segmentation model before counting them. To train this model, we propose a novel method to obtain pseudo-labeled segmentation masks. Specifically, we use an unsupervised image clustering method to generate a set of candidate pseudo object masks, from which we select the optimal one using a pretrained CAC model. We show that the trained segmentation model can effectively localize objects of interest based on the exemplars and prevent the model from counting everything. To properly evaluate the performance of CAC methods in real-world scenarios, we introduce two new benchmarks: a synthetic test set and a new test set of real images containing countable objects from multiple classes. Our proposed method shows a significant advantage over previous CAC methods on these two benchmarks.

1. Introduction

Class-agnostic counting (CAC) aims to infer the number of objects in an image, given a few object exemplars. Compared to conventional object counters that count objects from a specific category, *e.g.*, human crowds [32], cars [28], animals [3], or cells [39], CAC can count objects of an arbitrary category of interest, which enables numerous applications across various domains.

Most of the current CAC methods focus on capturing the intra-class similarity between image features [14,24,31,33]. For example, BMNet [33] adopts a self-similarity module to enhance the feature's robustness against intra-class variations. Another recent approach, SAFECount [43], uses a similarity-aware feature enhancement framework to better capture the support-query relationship. These methods perform quite well on the current benchmark, *i.e*. FSC-147, in which images only contain objects from a single dominant class. However, we observe that they often do not work well when there are distractors in the image and tend to count every single object regardless of the exemplars (Figure 1). This issue greatly limits the potential applicability of these methods in real-world scenarios. A possible reason is that the current counting datasets only contain single-class training images, causing the counting models to overlook the inter-class discriminability due to the absence of objects from different categories.

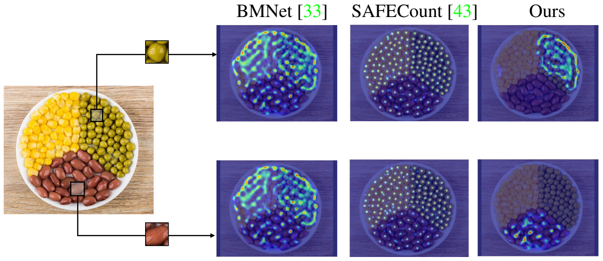


Figure 1. Visualizations of the density maps predicted by BMNet [33], SAFECount [43], and ours. BMNet and SAFECount count everything regardless of the annotated exemplars.

A natural solution to resolve this issue is to train the counting model with images containing objects of multiple classes. Since such training data does not exist, we synthesize them by concatenating multiple single-class images together. However, our experiments show that while the model trained on these images indeed performs better at discarding distractors, the counting performance drops significantly (5.3). This could be because selectively counting the objects of interest requires recognizing certain discriminative features that distinguish between different classes. However, this could decrease its robustness against the intra-class variations due to the invariance-discriminative power trade-off [35].

Therefore, we adopt a decoupled approach that employs two different networks for object localization and counting, respectively. Compared to existing methods that directly perform counting, we use an additional exemplar-based segmentation network to first localize the image patches containing objects of the correct category. Our main contribution lies in generating pseudo-labels for training this segmentation model. This is important since collecting large-scale data with mask annotations is both time-consuming and labor-intensive.

In this paper, we propose a novel method to obtain accurate pseudo-labeled segmentation masks. For each training image, we first generate a set of candidate semantic masks using an unsupervised image clustering method and then select from them the one that would yield the most accurate count when used together with a pre-trained object counter. We show that the segmentation model trained on those pseudo-labeled segmentation masks are significantly better than other alternative approaches of pseudo labelling such as similarity map or dot annotations (Table 3). In essence, we use the performance of a pre-trained object counter to pseudo-label the segmentation data. To the best of our knowledge, we are the first to employ this technique for obtaining pseudo-labeled segmentation masks.

Specifically, we use K-Means to obtain various candidate masks for masking the output similarity map of the counter. Each pixel in this map corresponds to a patch in the original image. These patches, along with provided exemplars, are represented by feature embeddings computed from a pre-trained ImageNet backbone [17]. We consider the patches whose embeddings fall into the same cluster as the exemplars to contain the objects of interest and assign positive labels to the corresponding mask pixels. We assign negative labels otherwise. The output of K-Means can vary significantly with different choice of K, depending on how many groups of irrelevant objects present in the image. In our case, we choose the pseudo mask that when using it to mask the output similarity map of the counting model, would produce the most accurate count.

To properly evaluate the performance of CAC methods in real-world scenarios, we introduce two new benchmarks, a synthetic dataset originating from FSC-147, and a new test set of real images in which objects from multiple classes are present. Our proposed method outperforms current counting methods by a large margin on these two benchmarks.

In short, our main contributions are:

- We identify a critical issue of the previous class-agnostic counting methods, *i.e.*, greedily counting every object when objects of multiple classes appear in the same image, and propose a simple segment-and-count strategy to resolve it.
- We propose a method to obtain pseudo-labeled segmentation masks using only annotated exemplars and use them to train a segmentation model.
- We introduce new benchmarks for class-agnostic object counting, on which our method outperforms the previous counting methods by a large margin.

2. Related Work

2.1. Class-specific Object Counting

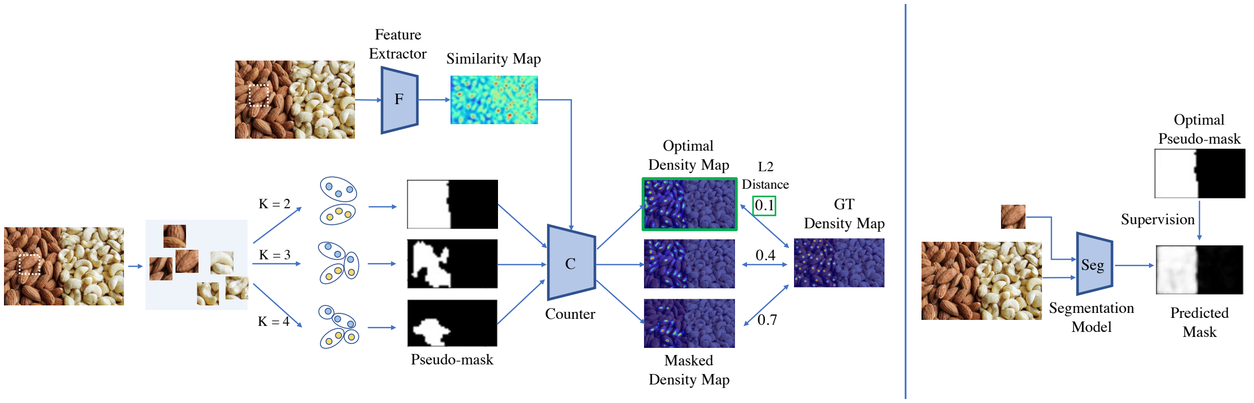
Class-specific object counting aims to count objects from pre-defined categories, such as humans [1, 20, 23, 25, 32, 34, 36, 38, 40, 45–48], animals [3], cells [39] and cars [18, 28]. Generally, there are two groups of class-specific counting methods: detection-based methods [6, 18, 22] and regression-based methods [5,9,10,26,37,46,48]. Detection-based methods apply an object detector on the image and count the number of objects based on the detected boxes. However, accurately detecting tiny objects still remains challenging [41]. Regression-based methods predict a density map for each input image, and the final result is obtained by summing up the pixel values. Both types of methods require a large amount of training data with rich training annotations. Moreover, they can not be used to count objects of arbitrary categories at test time.

2.2. Class-agnostic Object Counting

Class-agnostic object counting aims to count arbitrary categories given only a few exemplars [2, 14, 24, 27, 29, 31, 33, 42, 44]. Previous methods mostly focus on how to better capture the similarity between exemplars and image features. For example, SAFECount [43] uses a similarity-aware feature enhancement framework to better model the support-query relationship. RCAC [14] is proposed to enhance the counter's robustness against intra-class diversity. Nguyen *et al.* [29] recently introduce new benchmarks for object counting, which contains images of objects from multiple classes, originating from the FSC-147 and LVIS [15] datasets. However, these benchmarks are designed for the task of jointly detecting and counting object instances in complex scenes, where the central focus is on how to detect them accurately.

2.3. Unsupervised Semantic Segmentation

A closely related task to ours is unsupervised semantic segmentation [7, 8, 12, 13, 16, 19, 21, 30], which aims to discover classes of objects within images without external supervision. IIC [21] attempts to learn semantically meaningful features through transformation equivariance. PiCIE [8] further improves on IIC's segmentation results by incorporating geometric consistency as an inductive bias. Although these methods can semantically segment images without supervision, they typically require a large-scale dataset [4, 11] to learn an embedding space that is cluster-friendly. Moreover, the label space of semantic segmentation is limited to a set of pre-defined categories. In comparison, our goal is to localize the region of interest specified by a few exemplars, which can belong to an arbitrary class.



(a) Computing the optimal pseudo mask

(b) Training a segmentation model

Figure 2. Overview of our approach. We propose a method to obtain the pseudo segmentation masks using only box exemplars and dot annotations (a), and then use the obtained pseudo masks to train an exemplar-based segmentation model (b). Specifically, given an image and a few annotated exemplars, we crop a set of image patches, each of which corresponds to a mask pixel (we only visualize 6 patches here for simplicity). We run K-Means clustering on the feature embeddings extracted from all cropped patches and the exemplars. Those pixels whose embeddings fall into the same cluster as the exemplar form an object mask indicating the image area containing the objects of interest. We find the optimal number of clusters, K, such that the counting model can produce the density map closest to the ground truth after the pseudo mask is applied. We use the obtained pseudo masks to train an exemplar-based segmentation model, which can then be used to infer the object mask given an arbitrary test image.

3. Method

To resolve the counting-everything issue, we train a segmentation model to localize only objects to be counted. Figure 2 summarizes our approach. We propose a method to obtain pseudo segmentation masks using only box exemplars and dot annotations, and then use these pseudo masks to train an exemplar-based segmentation model.

Specifically, given an image and a few annotated exemplars, we tile the input image into different patches, each of which corresponds to a pixel on the mask. We run K-Means clustering on the feature embeddings extracted from all cropped patches and the exemplars. Those mask pixels whose corresponding patch embeddings fall into the same cluster as the exemplar will form an object mask indicating the image area containing the objects of interest. We find the optimal number of clusters, K, such that a pre-trained class-agnostic object counting model can produce the density map closest to the ground truth after the pseudo mask is applied. We use the obtained pseudo masks to train an exemplar-based segmentation model, which can then be used to infer the object mask given an arbitrary test image. For the rest of the paper, we denote the pre-trained single-class counting model as the “base counting model”. Below we will first describe how we train this base counting model and then present the detail of our proposed clustering-based pseudo mask acquisition method.

3.1. Training The Base Counting Model

We first train a base counting model using images from the single-class counting dataset [31]. Similar to previous works [31, 33], the base counting model uses the input image and the exemplars to obtain a density map for object counting. The model consists of a feature extractor F and a counter C. Given a query image I and an exemplar B of an arbitrary class c, we input I and B to the feature extractor to obtain the corresponding output, denoted as $F(I)$ and $F(B)$ respectively. $F(I)$ is a feature map of size $d * h_I * w_I$ and $F(B)$ is a feature map of size $d * h_B * w_B$. We further perform global average pooling on $F(B)$ to form a feature vector b of d dimensions.

After this feature extraction step, we obtain the similarity map S by correlating the exemplar feature vector b with the image feature map $F(I)$. Specifically, let $w_{(i,j)} = F_{(i,j)}(I)$ be the channel feature at spatial position (i, j), S can be computed by:

$$S_{(i,j)}(I, B) = w^T_{(i,j)} b. \tag{1}$$

In the case where n exemplars are given, we use Eq. 1 to calculate n similarity maps, and the final similarity map is the average of these n similarity maps.

We then concatenate the image feature map $F(I)$ with the similarity map S, and input them into the counter C to predict a density map D. The final predicted count N is obtained by summing over the predicted density map D:

$$N = \sum_{i,j} D_{(i,j)}, \tag{2}$$

where $D_{(i,j)}$ denotes the density value for pixel (i, j). The supervision signal for training the counting model is the L_2 loss between the predicted density map and the ground truth density map:

$$L_{\text{count}} = \|D(I, B) - D^*(I, B)\|_2^2, \tag{3}$$

where D^* denotes the ground truth density map.

3.2. Pseudo-Labeling Segmentation Masks

In this section, we describe our method to obtain pseudo-masks using only box exemplars and dot annotations. The mask is of the same size as the similarity map from the base counting model and each pixel on the mask is associated with a region in the original image. Ideally, the pixel value on the mask is 1 if the corresponding region contains the object of interest and 0 elsewhere. Specifically, for the pixel from the mask M at location (i, j), we find its corresponding patch $p(i, j)$ in the input image centering around (i_I, j_I), where $i_I = i * r + 0.5 * r$ and $j_I = j * r + 0.5 * r$. Here, r is the downsampling ratio between the original image and the similarity map. The width and height of $p(i, j)$ are set to be the mean of the width and height of the exemplar boxes.

We denote $\mathbb{P} = \{p_1, p_2, ...p_n\}$ as a set of image patches, each of which corresponds to one pixel in the mask. The goal is to assign a binary label to each patch indicating if it contains the object of interest or not. To achieve this, we first extract the ResNet-50 features for all patches in $\mathbb{P}$ to get a set of embeddings $\mathbb{F} = \{f_1, f_2, ...f_n\}$. Then we compute the average of the embeddings extracted from the exemplar boxes in this image, denoted as f_B. We run K-means on the union of $\{f_1, f_2, ...f_n\}$ and $\{f_B\}$. Those patches whose embeddings fall into the same cluster as f_B will be considered to contain the object of interest, and result in a 1 value in the corresponding pixel of the mask. On the contrary, the pixel value will be 0 if the corresponding patch embedding falls into a different cluster as f_B. Here K-Means groups similar objects together, which can serve our purpose of segmenting objects belonging to different classes.

It is worth noting that the number of clusters, denoted as K, has a large effect on the output binary mask and the final counting results. If K is too small, too many patch embeddings will fall into the same cluster as the exemplar embedding and the counter will over-count the objects; if K is set too high, too few embeddings will fall into the same cluster, which results in too many regions being masked out. In our case, we find the optimal K for each image that results in the binary mask that minimizes the counting error. Specifically, for an input image I and the annotated exemplar B, $S(I, B)$ denotes the similarity map outputted by the pre-trained counting model, and $M(I, B)^k$ denotes the mask obtained when the number of clusters is set to k. By applying $M(I, B)^k$ on $S(I, B)$, the similarity scores on the non-target area are set to a small constant value δ and the similarity scores on the target area remain the same:

$$S_{(i,j)}(I, B)^k = \begin{cases} S_{(i,j)}(I, B), & \text{if } M_{(i,j)}(I, B)^k = 1, \\ \delta, & \text{otherwise.} \end{cases} \tag{4}$$

We then input $S(I, B)^k$ to the pre-trained counter C to get the corresponding density map $D(I, B)^k$. We find the optimal k such that the L_2 loss between the predicted density map and the ground truth density map is the smallest:

$$k^* = \underset{k}{\operatorname{argmin}} \|D(I, B)^k - D^*(I)\|_2^2, \tag{5}$$

where k^* denotes the optimal k and $D^*(I)$ denotes the ground truth density map for input image I.

3.3. Training Exemplar-based Segmentation Model

After obtaining the optimal masks for all the images in the multi-class training set, we train a segmentation model P to predict the pseudo segmentation masks based on the input image and the corresponding exemplar. In particular, for an input image I and the annotated exemplar B, we first input I and B to the segmentation model to get the corresponding feature map output $P(I)$ and $P(B)$. We then apply global average pooling on $P(B)$ to form a feature vector v. In the case where multiple exemplars are provided, we apply global average pooling on each $P(B)$ and the final vector v is the average of all these pooling vectors.

The predicted mask M^p is obtained by computing the cosine similarity between v and the channel feature at each spatial location of $P(I)$. Specifically, the value of the predicted mask at position (i, j) is:

$$M_{(i,j)}(I, B)^p = \cos(P_{(i,j)}(I)^T, v). \tag{6}$$

The supervision signal for training this segmentation model is the L_2 loss between the predicted mask and the optimal mask obtained by finding the best k with Eq. 5:

$$L_{\text{seg}} = \|M(I, B)^p - M^*(I, B)\|_2^2, \tag{7}$$

where $M^*(I, B)$ denotes the optimal mask under k^*.

3.4. Inference on Testing Data

After the exemplar-based segmentation model is trained, we use it together with the pre-trained counting model to perform object counting. Given an input image for testing, we first input it to the feature extractor of the pre-trained counting model to get the corresponding similarity map. Then we use the segmentation model to predict a coarse mask where high values indicate the region of interets. We binarize this predicted mask with a simple threshold and apply it on the similarity map based on Eq. 4. The counter then take the masked similarity map as input and predict the density map and final object counts.

4. Experiments

4.1. Implementation Details

Network architecture For the base counting model, we use ResNet-50 as the backbone of the feature extractor, initialized with weights of a pre-trained ImageNet model. The backbone outputs feature maps of 1024 channels. For each query image, the number of channels is reduced to 256 using 1×1 convolution. For each exemplar, the feature maps are first processed with global average pooling and then linearly mapped to a 256-d feature vector. The counter consists of 5 convolution and bilinear upsampling layers to regress a density map of the same size as the query image. The segmentation model shares the same architecture as the backbone of the feature extractor. The output mask is of the same size as the similarity map from the base counting model.

Dataset We train the base counting model on the FSC-147 dataset. FSC-147 is the first large-scale dataset for class-agnostic counting. It includes 6135 images from 147 categories varying from animals, kitchen utensils, to vehicles. The categories in training, validation, and test sets have no overlap. We create synthetic images with 2 countable categories from FSC-147 dataset to train the segmentation model. Specifically, we randomly select two images belonging to different classes, crop a part from each image and then concatenate the two cropped parts horizontally. The synthetic validation set and test set contain 1431 and 1359 images respectively.

To properly evaluate the performance of class-agnostic counting in real-world scenarios, we further collect a test set of 450 images. Our test set includes objects from a wide range of categories, varying in scale and mixed together in diverse ways. For each image in this test set, there are at least two categories whose object instances appear multiple times. We provide dot annotations for 600 object instances groups. We test the trained model on both the synthetic test set and our collected real-world test set.

Training details Both the base counting model and the segmentation model are trained using the AdamW optimizer with a fixed learning rate of 10^{-5} and a batch size of 8. The base counting model is trained for 300 epochs and the segmentation model is trained for 20 epochs. We resize the input query image to a fixed height of 384, and the width is adjusted accordingly to preserve the aspect ratio of the original image. Exemplars are resized to 128×128 before being fed into the feature extractor. We run K-means on the extracted patch embeddings to find the K that leads to the optimal mask for each image. The embeddings are extracted from a pre-trained ImageNet backbone. The threshold for binarizing the segmentation mask is 0.6 and the number of clusters K ranges from 2 to 6.

4.2. Evaluation Metrics

For our collected real-world test set, the counting error ϵ for image i is defined as $\epsilon_i = |y_i - \hat{y}_i|$, where y_i and $\hat{y}_i$ are the ground truth and the predicted number of objects respectively. For our synthetic test set, the objects of interest are only present in the left / right-half part of the image. Ideally, the predicted number of objects should be close to the ground truth in the area of interest while being zero elsewhere. Thus, we define the counting error as $\epsilon_i = |y_i - \hat{y}_i| + \bar{\hat{y}}_i$, where $\hat{y}_i$ and $\bar{\hat{y}}_i$ denote the predicted number of objects in the interest area and non-interest area respectively.

We use Mean Average Error (MAE), Root Mean Squared Error (RMSE), Normalized Relative Error (NAE) and Squared Relative Error (SRE) to measure the performance of different object counters over all testing images. In particular, MAE = $\frac{1}{n}\sum_{i=1}^{n}\epsilon_i$; RMSE = $\sqrt{\frac{1}{n}\sum_{i=1}^{n}\epsilon_i^2}$; NAE = $\frac{1}{n}\sum_{i=1}^{n}\frac{\epsilon_i}{y_i}$; SRE = $\sqrt{\frac{1}{n}\sum_{i=1}^{n}\frac{\epsilon_i^2}{y_i}}$ where n is the number of testing images.

4.3. Comparing Methods

We compare our method with recent class-agnostic counting methods, including CounTR (Counting TRansformer [24]), FamNet (Few-shot adaptation and matching Network [31]), SAFECount (Similarity-Aware Feature Enhancement block for object Counting [43]) and BMNet (Bilinear Matching Network [33]).

4.4. Results

Quantitative results. We first use the synthetic images containing multiple categories to fine-tune existing class-agnostic counting methods, including FamNet (Few-shot adaptation and matching Network [31]), SAFECount (Similarity-Aware Feature Enhancement block for object Counting [43]) and BMNet (Bilinear Matching Network [33]). We denote the test-time adapted version of FamNet by FamNet+ following previous work [33]. Table 1 summarizes the results of these methods on the synthetic test set with and without fine-tuning. As shown in the table, fine-tuning improves all the methods. SAFECount, for example, shows an 8.30 error reduction w.r.t. MAE and an 11.99 error reduction w.r.t. RMSE on validation set. Interestingly, we find that FamNet, which has the largest counting error on the FSC-147 test set among these methods, performs best on the synthetic test dataset without fine-tuning. Unlike other methods, FamNet keeps the backbone of the counting model fixed during training, which prevents the model from overfitting into capturing the similarity between object instances and greedily counting everything. Furthermore,

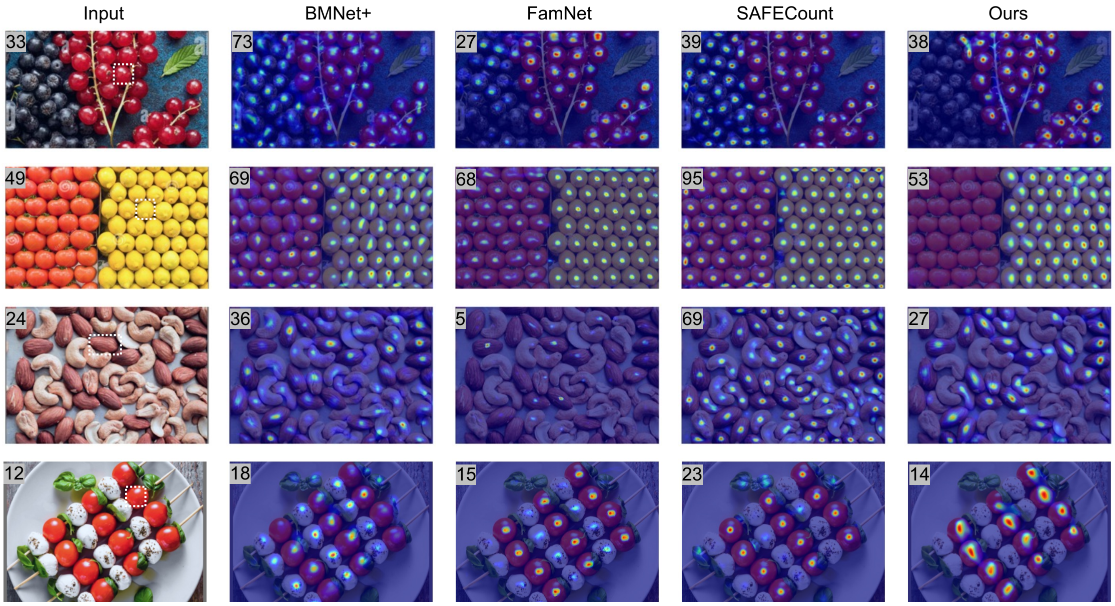


Figure 3. Qualitative results on our collected counting test dataset. We visualize a few input images, the corresponding annotated exemplar (bounded in a dashed white box) and the predicted density maps. Predicted object counts are shown at the top-left corner. Our predicted density maps can highlight the objects of interest specified by the annotated box, which will lead to more accurate object counts.

Method	Training Set	Synthetic Val Set				Synthetic Test Set			
		MAE	RMSE	NAE	SRE	MAE	RMSE	NAE	SRE
FamNet [31]	FSC-147	18.15	33.16	0.63	4.42	22.22	40.85	0.79	9.29
FamNet+ [31]		27.74	39.78	1.33	7.29	29.90	43.59	1.16	8.82
BMNet+ [33]		31.09	42.43	1.75	9.51	39.78	57.85	1.81	11.96
SAFECount [43]		22.57	34.81	1.20	7.49	26.40	40.60	1.13	9.45
FamNet [31]	FSC-147 + Synthetic	17.30	28.87	0.60	3.74	20.75	29.07	0.68	4.47
FamNet+ [31]		16.79	28.44	0.60	**3.71**	20.34	28.66	0.68	4.47
BMNet+ [33]		25.73	34.93	1.26	6.77	29.83	42.64	1.23	8.39
SAFECount [43]		**14.27**	**22.82**	**0.59**	3.89	15.79	34.16	0.65	8.25
Seg-then-Count	-	14.34	26.03	0.61	4.48	**11.13**	**16.96**	**0.41**	**2.80**

Table 1. Performance on our synthetic test set. Our proposed 'Seg-then-Count' strategy effectively alleviates the counting-everything issue.

we present the results of our proposed method, "seg-then-count", which utilizes an additional segmentation model tailored specifically for segmentation tasks. As shown in the table, our proposed method achieves the lowest MAE and RMSE on the test set.

Table 2 shows the results on our collected real-world test set. Similarly, both fine-tuning existing counting models and training another segmentation model effectively alleviates the counting-everything issue. Segment-then-count generalizes better to real-world images, achieving an error rate of 6.97 in terms of MAE and 13.03 in terms of RMSE.

Qualitative analysis. In Figure 3, we present a few input testing images, the corresponding annotated bounding box and the density maps produced by different counting methods. We can see that when there are objects of multiple classes present in the image, previous methods fail to distinguish them accurately, which often leads to over-counting. In comparison, the density map predicted by our method can highlight the objects of interest specified by the annotated box, even for the hard case where the objects are irregularly placed in the image (the 3rd row).

Method	Training Set	Real Set	
		Test MAE	Test RMSE
FamNet [31]	FSC-147	13.03	20.28
FamNet+ [31]		19.42	39.78
BMNet+ [33]		25.55	40.35
SAFECount [43]		23.57	40.99
FamNet [31]	FSC-147 + Synthetic	11.03	17.60
FamNet+ [31]		11.31	18.84
BMNet+ [33]		11.44	23.22
SAFECount [43]		9.80	32.40
Seg-then-Count	-	**6.97**	**13.03**

Table 2. Performance of fine-tuning existing CAC models and using another segmentation model on the real-world test set. 'Seg-then-Count' generalizes better to real-world images containing multiple countable categories.

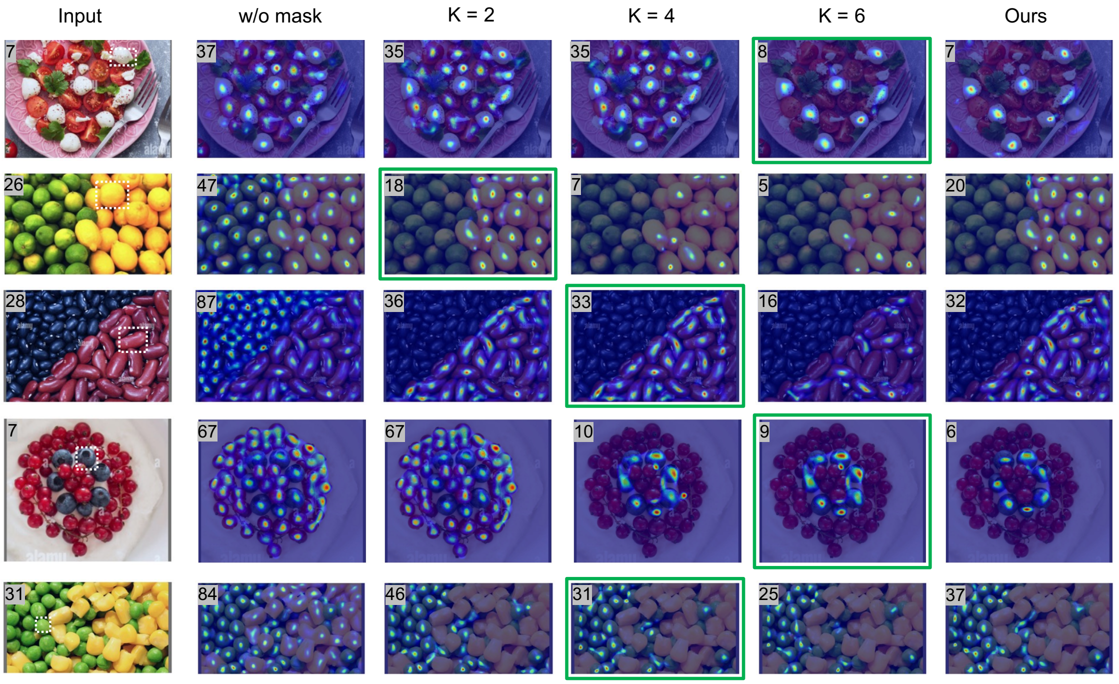


Figure 4. Qualitative analysis on the number of clusters. We visualize a few input images, the corresponding annotated exemplar (bounded in a dashed white box) and the density maps when using masks computed from K-means as well as predicted by our segmentation model. Predicted counting results are shown at the top-left corner. The density maps under the optimal K are framed in green. The value of K has a large effect on the counting results and the optimal K varies from image to image.

5. Analysis

5.1. Comparison of Pseudo-labeling Methods

In this section, we compare our proposed clustering-based pseudo-labeling method with other pseudo-labeling methods including 1) binarizing the similarity map between the image and the exemplar, and 2) creating pseudo boxes from dot annotations. For the first approach, we use a pre-trained feature extractor (ResNet-50 pre-trained on ImageNet) to extract the feature maps from the image and the exemplar. Then we correlate the pooled exemplar feature with the image feature to get the similarity map. The pseudo mask is obtained by binarizing this similarity map with a threshold. For the second approach, we create a pseudo box centering around each annotated dot. The size of the box is the average of the annotated exemplars. These pseudo boxes form a mask containing all the object dots. Results of these two approaches are summarized in Table 3. As can be seen, our pseudo masks result in most accurate object counts, outperforming other methods consistently.

5.2. Analysis on the Number of Clusters

When running K-means, the number of clusters, K, has a large effect on the computed binary mask and the final counting results. However, it is non-trivial to determine K given an arbitrary image. To resolve this issue, we first compute the optimal pseudo masks for the training images based on the dot annotations. Then we train an exemplar-based segmentation model to predict the obtained pseudo masks. During testing, we can use the trained model to predict the segmentation mask based on exemplars. In this section, we provide analyses on how K affects the final counting results and show a comparison with our proposed method.

Pseudo Masks	Threshold	Synthetic		Real	
		Val MAE	Test MAE	MAE	RMSE
w/o Mask	-	32.46	42.22	24.68	41.70
Similarity Map	0.2	31.35	38.63	24.94	37.60
	0.4	20.91	22.95	11.08	19.78
	0.6	27.12	27.52	17.93	29.76
	0.8	30.50	32.60	20.67	31.85
Dot Annotation	-	18.93	12.48	9.26	19.23
K-Means	-	**14.34**	**11.13**	**6.97**	**13.03**

Table 3. Comparison with pseudo-labeling via binarizing the similarity map and creating pseudo boxes from dot annotations. Our proposed method consistently outperforms other pseudo-labeling methods.

5.2.1 Quantitative Results

We report the counting performance when computing masks by running K-means under different values of K as well as using our predicted masks on the collected test set. Results are summarized in Table 4. As K goes from 2 to 6, both the MAE and RMSE decrease first and then increase, achieving the lowest when $K = 5$, *i.e.*, 7.98 w.r.t. MAE and 14.93 w.r.t. RMSE. Using our predicted masks outperforms the performance under the best K by 12.6% w.r.t. MAE and 12.7% w.r.t. RMSE, which demonstrates the advantages of using our trained segmentation model to predict the mask.

5.2.2 Qualitative Results

In Figure 4, we visualize a few input images and the corresponding density maps when using masks computed from K-means as well as using masks predicted by our segmentation model. The choice of K has a large effect on the counting results. If K is too small, too many patch embeddings will fall into the same cluster as the exemplar embedding and the counter will over-count the objects (the 4th row when K = 2); if K is too large, too few embeddings will fall into the same cluster, which results in too many regions being masked out (the 2nd row when K = 6). The optimal K varies from image to image, and it is non-trivial to determine the optimal K for an arbitrary image. Using our trained segmentation model, on the other hand, does not require any prior knowledge about the test image while producing more accurate masks and density maps based on the provided exemplars.

K	2	3	4	5	6	Ours
MAE	15.13	10.77	8.17	7.98	8.03	**6.97**
RMSE	28.09	20.71	15.38	14.93	15.31	**13.03**
NAE	0.94	0.63	0.44	0.42	0.40	**0.37**
SRE	1.68	1.18	0.69	0.62	**0.54**	**0.54**

Table 4. Quantitative analysis on the number of clusters. Our proposed method outperforms K-Means under different values of K on our collected test set.

5.3. Analysis on the Trade-off between Invariance and Discriminative Power

We observe that fine-tuning existing CAC models with synthetic data containing multiple categories will negatively impact their counting ability. Our explanation is that when images contain objects from a single dominant class, the model will focus on capturing the similarity between instances to minimize the counting errors, while ignoring the discrepancy between categories; when images contain objects from multiple classes, the model will instead focus more on the inter-class discrepancy to distinguish between them. To get a better understanding of this trade-off, we provide the detailed feature distribution statistics in Table 5. Specifically, we measure the intra-class distance and inter-class distance of the exemplar features extracted from the counting model before and after fine-tuning on the synthetic dataset. Intra-class distance refers to the mean of Euclidean distance between a feature embedding and the corresponding class's embedding center. Inter-class distance refers to the mean of the minimum distance between embedding centers. As shown in the table, after fine-tuning the model using the synthetic images, both intra-class distance and inter-class distance increase. Larger inter-class distance means features from different classes are more separable, suggesting a increased discriminative power of the model; larger intra-class distance means features within the same class are less compact, suggesting decreased robustness against within-class variations. This trade-off between invariance and discriminative power makes it challenging for one model to distinguish and count simultaneously. Correspondingly, we observe an error increase on the FSC-147 test set and an error decrease on the synthetic test set after fine-tuning.

Split	Training Set	Intra	Inter	FSC-147 MAE	Synthetic MAE
Val	FSC-147	2.35	1.12	18.55	32.46
	FSC-147+Synthetic	2.90	1.30	32.36 (+13.81)	25.74 (-6.99)
Test	FSC-147	2.31	1.19	20.68	42.22
	FSC-147+Synthetic	2.86	1.48	32.34 (+11.66)	29.12 (-13.1)

Table 5. Analysis on the trade-off between invariance and discriminative power of the counting model. After fine-tuning on the synthetic dataset, both the intra-class and inter-class distances of exemplar features become larger, leading to an error increase on FSC-147 test set and an error decrease on the synthetic test set.

6. Conclusion

In this paper, we identify a critical issue of the previous class-agnostic counting methods, *i.e.*, greedily counting every object when objects of multiple classes appear in the same image. We show that simply training the counting model with synthetic data can alleviate this issue but often at the price of sacrificing the ability to count objects from a single class accurately. Thus, our strategy is to localize the area of interest first and then count the objects inside the area. To do this, we propose a method to obtain pseudo segmentation masks using only box exemplars and dot annotations. We show that a segmentation model trained with these pseudo-labeled masks can effectively localize the image area containing the objects of interest for an arbitrary testing image.

References

[1] Shahira Abousamra, Minh Hoai, Dimitris Samaras, and Chao Chen. Localization in the crowd with topological constraints. *arXiv*, 2021. 2

[2] Carlos Arteta, Victor S. Lempitsky, Julia Alison Noble, and Andrew Zisserman. Interactive object counting. In *European Conference on Computer Vision (ECCV)*, 2014. 2

[3] Carlos Arteta, Victor S. Lempitsky, and Andrew Zisserman. Counting in the wild. In *European Conference on Computer Vision (ECCV)*, 2016. 1, 2

[4] Holger Caesar, Jasper R. R. Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[5] Antoni B. Chan, Zhang-Sheng John Liang, and Nuno Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 2

[6] Prithvijit Chattopadhyay, Ramakrishna Vedantam, Ramprasaath R. Selvaraju, Dhruv Batra, and Devi Parikh. Counting everyday objects in everyday scenes. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[7] Mickaël Chen, Thierry Artières, and Ludovic Denoyer. Unsupervised object segmentation by redrawing. In *Proceedings of the Neural Information Processing Systems (NeurIPS)*, 2019. 2

[8] Jang Hyun Cho, Utkarsh Mall, Kavita Bala, and Bharath Hariharan. Picie: Unsupervised semantic segmentation using invariance and equivariance in clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[9] Hisham Cholakkal, Guolei Sun, Fahad Shahbaz Khan, and Ling Shao. Object counting and instance segmentation with image-level supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[10] Hisham Cholakkal, Guolei Sun, Salman Hameed Khan, Fahad Shahbaz Khan, Ling Shao, and Luc Van Gool. Towards partial supervision for generic object counting in natural scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[11] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[12] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Unsupervised semantic segmentation by contrasting object mask proposals. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[13] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2

[14] Shenjian Gong, Shanshan Zhang, Jian Yang, Dengxin Dai, and Bernt Schiele. Class-agnostic object counting robust to intraclass diversity. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 1, 2

[15] Agrim Gupta, Piotr Dollár, and Ross B. Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[16] Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T. Freeman. Unsupervised semantic segmentation by distilling feature correspondences. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. 2

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[18] Meng-Ru Hsieh, Yen-Liang Lin, and Winston H. Hsu. Drone-based object counting by spatially regularized regional proposal network. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[19] Jyh-Jing Hwang, Stella X. Yu, Jianbo Shi, Maxwell D. Collins, Tien-Ju Yang, Xiao Zhang, and Liang-Chieh Chen. Segsort: Segmentation by discriminative sorting of segments. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[20] Haroon Idrees, Muhmmad Tayyab, Kishan Athrey, Dong Zhang, Somaya Ali Al-Maadeed, Nasir M. Rajpoot, and Mubarak Shah. Composition loss for counting, density map estimation and localization in dense crowds. In *European Conference on Computer Vision (ECCV)*, 2018. 2

[21] Xu Ji, Joao F. Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[22] Issam H. Laradji, Negar Rostamzadeh, Pedro H. O. Pinheiro, David Vázquez, and Mark W. Schmidt. Where are the blobs: Counting by localization with point supervision. *arXiv*, 2018. 2

[23] Dongze Lian, Jing Li, Jia Zheng, Weixin Luo, and Shenghua Gao. Density map regression guided detection network for rgb-d crowd counting and localization. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[24] Chang Liu, Yujie Zhong, Andrew Zisserman, and Weidi Xie. Countr: Transformer-based generalised visual counting. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2022. 1, 2, 5

[25] Weizhe Liu, N. Durasov, and P. Fua. Leveraging self-supervision for cross-domain crowd counting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[26] Weizhe Liu, Mathieu Salzmann, and Pascal V. Fua. Context-aware crowd counting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[27] Erika Lu, Weidi Xie, and Andrew Zisserman. Class-agnostic counting. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2018. 2

[28] Terrell N. Mundhenk, Goran Konjevod, Wesam A. Sakla, and Kofi Boakye. A large contextual dataset for classification, detection and counting of cars with deep learning. In *European Conference on Computer Vision (ECCV)*, 2016. 1, 2

[29] Thanh Nguyen, Chau Pham, Khoi Nguyen, and Minh Hoai. Few-shot object counting and detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 2

[30] Yassine Ouali, Céline Hudelot, and Myriam Tami. Autoregressive unsupervised image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2

[31] Viresh Ranjan, Udbhav Sharma, Thua Nguyen, and Minh Hoai. Learning to count everything. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2, 3, 5, 6, 7

[32] Deepak Babu Sam, Abhinav Agarwalla, Jimmy Joseph, Vishwanath A. Sindagi, R. Venkatesh Babu, and Vishal M. Patel. Completely self-supervised crowd counting via distribution matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 1, 2

[33] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2, 3, 5, 6, 7

[34] Vishwanath A. Sindagi, Rajeev Yasarla, and Vishal M. Patel. Pushing the frontiers of unconstrained crowd counting: New dataset and benchmark method. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[35] Manik Varma and Debajyoti Ray. Learning the discriminative power-invariance trade-off. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2007. 1

[36] Jia Wan, Ziquan Liu, and Antoni B. Chan. A generalized loss function for crowd counting and localization. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[37] Boyu Wang, Huidong Liu abd Dimitris Samaras, and Minh Hoai Nguyen. Distribution matching for crowd counting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2

[38] Qi Wang, Junyu Gao, Wei Lin, and Xuelong Li. Nwpu-crowd: A large-scale benchmark for crowd counting and localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43:2141–2149, 2021. 2

[39] Weidi Xie, J. Alison Noble, and Andrew Zisserman. Microscopy cell counting and detection with fully convolutional regression networks. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, 2018. 1, 2

[40] Haipeng Xiong and Angela Yao. Discrete-constrained regression for local counting models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 2

[41] Cheng Xu, Yifan Li, Yu Xie, and Jianbo Wang. Dynamic coarse-to-fine learning for oriented tiny object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[42] Shuo Yang, Hung-Ting Su, Winston H. Hsu, and Wen-Chin Chen. Class-agnostic few-shot object counting. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, 2021. 2

[43] Zhiyuan You, Yujun Shen, Kai Yang, Wenhan Luo, X. Lu, Lei Cui, and Xinyi Le. Few-shot object counting with similarity-aware feature enhancement. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, 2023. 1, 2, 5, 6, 7

[44] Zhiyuan You, Kai Yang, Wenhan Luo, Xin Lu, Lei Cui, and Xinyi Le. Few-shot object counting with similarity-aware feature enhancement. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, 2023. 2

[45] Anran Zhang, Lei Yue, Jiayi Shen, Fan Zhu, Xiantong Zhen, Xianbin Cao, and Ling Shao. Attentional neural fields for crowd counting. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[46] Cong Zhang, Hongsheng Li, Xiaogang Wang, and Xiaokang Yang. Cross-scene crowd counting via deep convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[47] Qi Zhang and Antoni Chan. Calibration-free multi-view crowd counting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 2

[48] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2