Learning to Count from Pseudo-Labeled Segmentation Supplementary Material

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

1. Overview

In this document, we provide additional experiments and analyses. In particular:

- In Section 2, we report the full results of different counting methods on both the FSC-147 test set and our synthetic test set.
- In Section 3, we report the results on the CARPK dataset.
- In Section 4, we report the performance of using different loss functions for training the segmentation model.
- In Section 5, we provide analysis on the patch size.
- In Section 6, we compare our proposed method with two alternative approaches to obtain pseudo masks.
- In Section 7, we report the time cost of running K-Means clustering at test time and using our trained segmentation model.
- In Section 8, we provide additional qualitative analysis on the number of clusters.
- In Section 9, we provide additional qualitative comparisons of different counting methods on our collected real-world test images.
- In Section 10, we provide more details about our collected test set.

2. Results on FSC-147 and Synthetic Test Set

In this section, we report the results of different class-agnostic counting methods on both the FSC-147 dataset and synthetic test set with and without fine-tuning. As shown in Table 1, fine-tuning using the synthetic composite images improves the performance of all methods on the synthetic test set by a large margin. FamNet+ [2], for example, shows a 10.95 error reduction w.r.t. mean absolute error (MAE) on the validation set and and a 9.56 error reduction w.r.t. MAE on the test set. At the mean time, the performance of all methods on FSC-147 test set drops after fine-tuning. As discussed in the main paper, learning a single model to distinguish and count simultaneously is challenging, leading to the observed performance trade-off. In addition, we present the results of using an additional model specifically for segmentation (denoted as 'seg-then-count'), which has an error rate of 14.34 on the validation set and error rate of 11.13 on the test set.

3. Results on CARPK

In this section, we test our models's generality on a car counting dataset CARPK [1] following previous methods [2, 3]. CARPK contains 1,448 images of parking lots in a bird view, which differs significantly from the images in FSC147. As shown in Tab. 2, our method slightly outperforms BMNet [3] and SAFECount [4] on CARPK. We note that our approach is designed to mask out distracting objects, which is generally not a significant issue in CARPK. Unlike FSC-147, which includes 147 categories, CARPK focuses solely on counting cars, which are easy to distinguish from potential distractors like trees and people. Therefore, our segment-and-count method does not show significant improvements in this context. We have included the results in the revision.

Method	Training Set	FSC-147 Val MAE	FSC-147 Test MAE	Synthetic Val MAE	Synthetic Test MAE
FamNet [2]	FSC-147	24.32	22.56	18.15	22.22
	FSC-147+Synthetic	30.88 (+ 6.56)	28.40 (+ 5.84)	17.30 (- 0.85)	20.75 (- 1.47)
FamNet+ [2]	FSC-147	23.75	22.08	27.74	29.90
	FSC-147+Synthetic	29.45 (+ 5.70)	26.93 (+ 4.85)	16.79 (- 10.95)	20.34 (- 9.56)
BMNet+ [3]	FSC-147	15.74	14.62	31.09	39.78
	FSC-147+Synthetic	24.24 (+ 8.50)	20.89 (+ 6.27)	25.73 (- 5.36)	29.83 (- 9.95)
SAFECount [4]	FSC-147	14.42	13.56	22.57	26.40
	FSC-147+Synthetic	27.65 (+ 13.23)	27.24 (+ 13.68)	14.27 (- 8.30)	15.79 (- 10.61)
Seg-then-Count	-	18.55	20.68	14.34	11.13

Table 1. Results of different methods on FSC-147 test set and our synthetic test set. Both fine-tuning the existing models and training an additional model for segmentation (denoted as 'seg-then-count') effectively alleviates the counting-everything issue.

	FamNet [2]	BMNet [3]	BMNet+ [3]	SAFECount [4]	Ours
MAE	18.19	8.05	5.76	4.91	**4.74**
RMSE	33.66	9.70	7.83	6.32	**6.29**

Table 2. Performance on the CARPK dataset.

4. Ablation on Loss Functions for the Segmentation Model

We choose L2 loss for training the segmentation model in the main experiments. In this section, we experiment with another commonly used loss function, cross-entropy loss and report the results in Table 3. We observe comparable performance between the two loss functions.

Loss Function	Synthetic Test Val MAE	Synthetic Test Val RMSE	Synthetic Test Test MAE	Synthetic Test Test RMSE	Real Test MAE	Real Test RMSE
Cross-entropy	14.11	26.62	11.39	16.74	7.38	13.31
L2	14.34	26.03	11.13	16.96	6.97	13.03

Table 3. Performance of using cross-entropy loss and L2 loss.

5. Ablation on Patch Size

When computing the pseudo masks, each pixel on the mask is associated with a a region in the original image. In our main experiments, we use the mean size of the annotated exemplars. In this section, we explore alternative configurations, including the minimum and maximum sizes of the annotated exemplars. The results are summarized in Table **??**. We observe that using the mean size yields slightly better performance.

Patch Size	Synthetic Test Val MAE	Synthetic Test Val RMSE	Synthetic Test Test MAE	Synthetic Test Test RMSE	Real Test MAE	Real Test RMSE
Min	14.93	27.11	11.74	17.25	7.01	13.18
Max	15.37	28.58	12.06	17.44	7.33	13.29
Mean	14.34	26.03	11.13	16.96	6.97	13.03

Table 4. Analysis on the patch size.

6. Ablation on Pseudo-labeling Method

In Section 5.1 of the main manuscript, we compare our proposed clustering-based pseudo-labeling method with two other pseudo-labeling methods. Parts of the results were not included due to space constraints. We present the complete comparison results in this section, including the root mean squared error (RMSE) of all methods and the results of using pseudo masks from the dot annotations with different box sizes.

6.1. Comparing with Pseudo-labeling via Binarizing Similarity Maps

We first compare our proposed method with pseudo-labeling via binarizing the similarity map between the image and the exemplar. Specifically, we use a pre-trained feature extractor 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. We experiment with different thresholds and the results are summarized in Table 5. We observe that the threshold for binarizing similarity maps has a large impact on the final performance. When the threshold is set to 0.4, the error rate achieves the lowest on the synthetic set, *i.e.*, an MAE of 20.91 on the validation set and 22.95 on the test set. Our proposed method outperforms binarizing similarity maps by a large margin, achieving an MAE of 6.97 on the real-world test set.

Pseudo Masks	Threshold	Synthetic Test				Real Test	
		Val MAE	Val RMSE	Test MAE	Test RMSE	MAE	RMSE
w/o Mask	-	32.46	45.25	42.22	59.95	24.68	41.70
Similarity Map	0.2	31.35	41.90	38.63	53.00	24.94	37.60
	0.4	20.91	34.18	22.95	32.74	11.08	19.78
	0.6	27.12	44.88	27.52	40.09	17.93	29.76
	0.8	30.50	47.79	32.60	44.07	20.67	31.85
K-Means Clustering	-	**14.34**	**26.03**	**11.13**	**16.96**	**6.97**	**13.03**

Table 5. Comparison with pseudo-labeling via binarizing the similarity map between the image and the exemplar. Our proposed method consistently outperforms binarizing similarity maps using different thresholds.

6.2. Comparing with Pseudo-labeling from Dot Annotations

An alternative way to obtain pseudo-labeled data for training the segmentation model is to create pseudo boxes from dot annotations. Specifically, we create a pseudo box centering around each annotated dot. These pseudo boxes form a mask containing all the object dots. We experiment with three different sizes of pseudo box, *i.e.*, the mean, minimum and maximum size of all exemplars. The performance of the model trained on these masks is shown in Table 6. Our proposed method outperforms pseudo-labeling with dot annotations on both the synthetic test set and our collected real-world test set consistently. On our collected test set, for example, the lowest MAE by pseudo-labeling from dot annotations is 8.57, which is a 22.0% error increase compared with our method. The results validate the advantage of our proposed method over pseudo-labeling using dot annotations.

Pseudo Masks	Box Size	Synthetic Test				Real Test	
		Val MAE	Val RMSE	Test MAE	Test RMSE	MAE	RMSE
w/o Mask	-	32.46	45.25	42.22	59.95	24.68	41.70
Dot Annotation	Mean	18.93	35.30	12.48	21.51	9.26	19.23
	Min	18.46	34.08	13.67	23.02	8.57	16.67
	Max	20.10	39.82	13.76	23.77	8.73	16.97
K-Means Clustering	-	**14.34**	**26.03**	**11.13**	**16.96**	**6.97**	**13.03**

Table 6. Comparison with pseudo-labeling from dot annotations. Our proposed method achieves lower counting errors on both the synthetic test set and our collected real-world test set.

7. Inference Time Comparison

In Section 5.2 of the main manuscript, we show that using the trained segmentation model to get object masks consistently outperforms running K-Means at test time. In this section, we compare the inference time of these two approaches. The average time costs (per image in second) are summarized in Table 7. As shown in the table, running K-Means results in a significantly higher time consumption. As the value of K increases, the time consumption increases accordingly. In comparison, our trained segmentation model only results in marginal additional computation time, *i.e.*, 0.015s per real test image and 0.012s per synthetic image. Using our trained model to obtain the objects masks is much faster than running K-Means at test time.

8. Qualitative Analysis on the Number of Clusters

In this section, we provide additional qualitative analysis on how the number of clusters, K, affects the final counting results. As shown in Figure 1, we visualize a few input images and the corresponding density maps when using masks

Test Set	w/o mask	K-Means $k=2$	$k=3$	$k=4$	$k=5$	$k=6$	Segmentation Model
Real-world set	0.047	0.722	0.848	0.970	1.069	1.161	0.061
Synthetic set	0.021	0.767	0.862	0.924	1.012	1.061	0.033

Table 7. The average time cost of running K-Means and using our segmentation model on the collected test set and our synthetic test set. All results are in the unit of seconds. Our proposed method only takes around 30 to 60 ms, which is much faster than K-Means.

computed from K-Means and using masks predicted by our segmentation model. The choice of K has a large effect on the counting results: a small K might lead to over-counting while a large K might cause objects of interest to be masked out. The optimal K varies from image to image, and it is non-trivial to determine the optimal K for an arbitrary image. Instead, using our trained segmentation model consistently produces accurate object masks and density maps based on the provided exemplars.

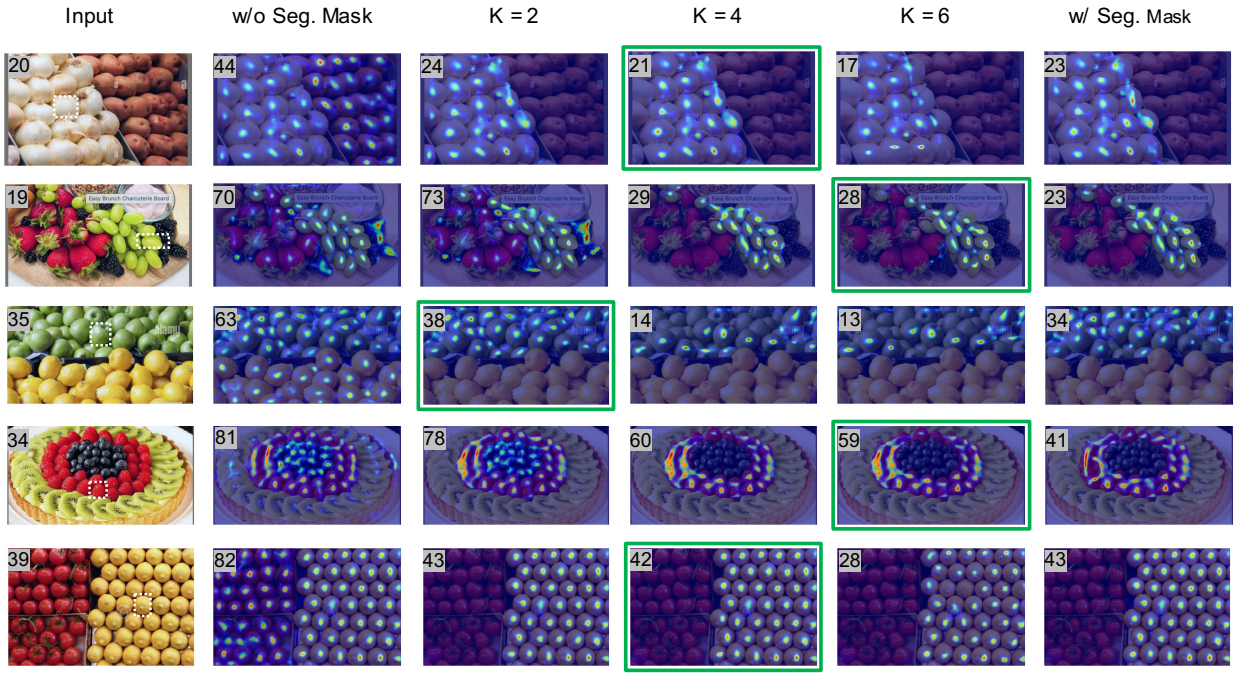

Figure 1. Qualitative analysis on the number of clusters. We visualize a few input images together with the corresponding annotated exemplar (bounded in a dashed white box) and the density maps when using masks computed from K-Means and masks predicted by our segmentation model. We only visualize one exemplar per image here for simplicity. Predicted counting results are shown in 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 value of K varies from image to image.

9. Qualitative Results

In this section, we provide additional qualitative results of using our trained segmentation model for class-agnostic object counting. In Figure 3, we present a few input testing images, the corresponding annotated bounding box and the density maps produced by different counting methods. As can be seen from the figure, 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 after applying our segmentation model highlights the objects of interest specified by the annotated box.

10. Details on the Collected Real-world Test Set

Although the current dataset for class-agnostic counting, FSC-147 [2], contains a large number of images with various object instances, the objects within each image are mostly from a single dominant class. However, in practice, there can be

Figure 2. Sample images from FSC-147 dataset and our collected real-world test set. (a). Images from FSC-147 mostly contain objects from a single dominant class. (b). Images from our collected test set contain objects from multiple classes.

objects from multiple classes in the image, which is more challenging since the counter needs to selectively count only the objects of interest. To evaluate the performance of different methods in this practical scenario, we collect and annotate a new test set of 450 images, in which objects from different categories are present. 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 groups of object instances. For each group, we randomly select 1 to 3 object instances as exemplar instances and annotate them with bounding boxes. Some sample images are shown in Figure 2. Our test set includes objects of different categories mixed together in various ways, including overlapping, adjacent placement, and random distribution. In addition, the objects vary in scale, ranging from small items like beans and peas to larger items like apples and watermelons. Overall, 76 out of 450 images contain objects of varying scales. We observe that the model's performance on these images is comparable to its performance across the entire test set.

References

[1] Meng-Ru Hsieh, Yen-Liang Lin, and Winston H. Hsu. Drone-based object counting by spatially regularized regional proposal network. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 1

[2] 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, 4

[3] 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

[4] 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

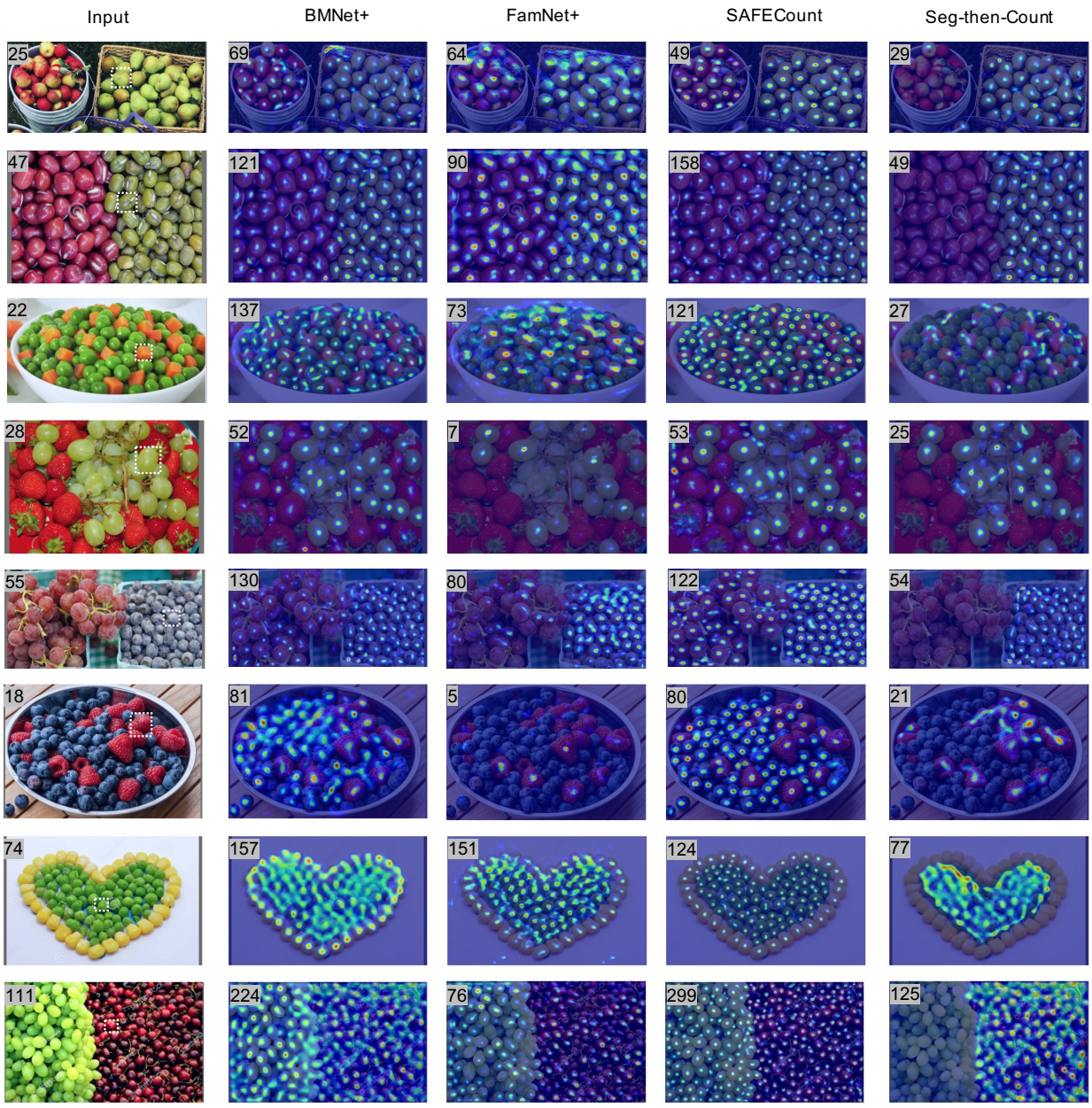

Figure 3. Qualitative results on our collected real-world test set. We visualize a few input images, the corresponding annotated exemplar (bounded in a dashed white box) and the predicted density maps. Predicted object counts are shown in the top-left corner. Using our trained segmentation model, the predicted density maps highlight the objects of interest specified by the annotated box, which leads to more accurate object counts.