Represent, Compare, and Learn: A Similarity-Aware Framework for Class-Agnostic Counting

Supplementary Material

Min Shi Hao Lu Chen Feng Chengxin Liu Zhiguo Cao
Key Laboratory of Image Processing and Intelligent Control, Ministry of Education
School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, China
{min_shi, hlu, chen_feng, cx_liu, zgcao}@hust.edu.cn

1. Introduction

This supplementary material includes the following contents:

- Detailed network architectures of BMNet and BMNet+.
- Discussion on why decomposing the original bilinear metric with one transformation matrix into two separate ones for exemplar and query (Sec. 3.1, page 3).
- More visualizations of dynamic channel attention weights for different categories (Sec. 3.2, page 4).
- More visualizations of density maps and counting results predicted by our methods and the previous state-of-the-art (Sec. 4.1, page 6).
- Qualitative evidence to show the benefit of combing query features with similarity maps for final counting regression (Sec. 4.4, page 8).
- The advantage of self-similarity module under few-shot scenarios (Sec. 4.3, page 8).

Note that the experiments in this material are conducted based on FSC147 dataset [3]. For clarity, we use notations in the main body of the paper: $\boldsymbol{z} = F(Z)$ for the exemplar feature, $\boldsymbol{x}_{ij} = F_{ij}(X)$ for the channel feature in query feature maps $F(X)$ at spatial position (i, j).

2. Network Architecture

Here we detail the modules within our network. Our code is at https://tiny.one/BMNet

2.1. Feature extractor

The configuration of feature extractor is shown in Fig. 1. We use the first four blocks of ResNet-50 [1] as the shared backbone for exemplar and query image as in FamNet [3]. The backbone outputs downsampled feature map whose size is $1/16$ of the original query image. Then, for the feature maps of each query image, we add an 1×1 convolution layer to reduce its channels from 1024 to 256. For the feature maps of each exemplar, they are first processed with global average pooling and then mapped via a single linear layer into a 256-dimensional feature vector.

2.2. Self-similarity Module

The configuration of self-similarity module is shown in Fig. 2. Both the input and output of this module are a sequence of feature vectors w.r.t. the exemplar and query image. Suppose we have n exemplars yielding $F(Z) \in \mathbb{R}^{n \times d}$ and one query image yielding $F(X) \in \mathbb{R}^{d \times h \times w}$. In this case, the input sequence IS is a feature set of d-dimensional exemplar features and query features, whose cardinality is $hw + n$. First, scale embedding is injected into IS, which yields a sequence denoted by E_{IS} (refer to Sec. 3.4 in the main paper for details). Then following the standard pipeline in self-attention [5], E_{IS} is projected to query Q_{ss}, key K_{ss}, and value V_{ss} with three matrices W_q, W_k, and W_v. Here the subscript ss denotes self-similarity. Then we obtain self-similarity matrix $A = \mathrm{softmax}(Q_{ss}^T K_{ss})$ where $A \in \mathbb{R}^{(hw+n) \times (hw+n)}$. A helps aggregate the features in value V_{ss} via $V_{ss}A$. Afterwards, the sequence $V_{ss}A$ is added back into the original input sequence with a

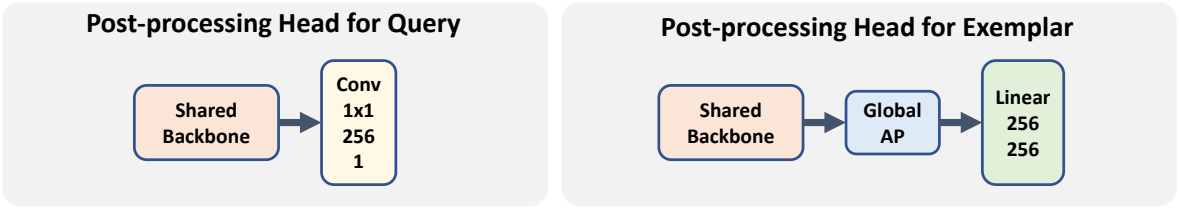

Figure 1. **Feature extractor in BMNet and BMNet+.** The exemplar and query image are processed with the same backbone as in FamNet [3] but with two different post-processing heads. The ReLU activation after each convolution layer is omitted for simplification. Global AP denotes global average pooling. Conv denotes convolutional layer, whose parameters stand for kernel size (the 1st row), output channel dimension (the 2nd row), and stride (the 3rd row). Linear denotes fully connected layer, whose parameters stand for input dimension (the 1st row) and output dimension (the 2nd row). The same hereinafter.

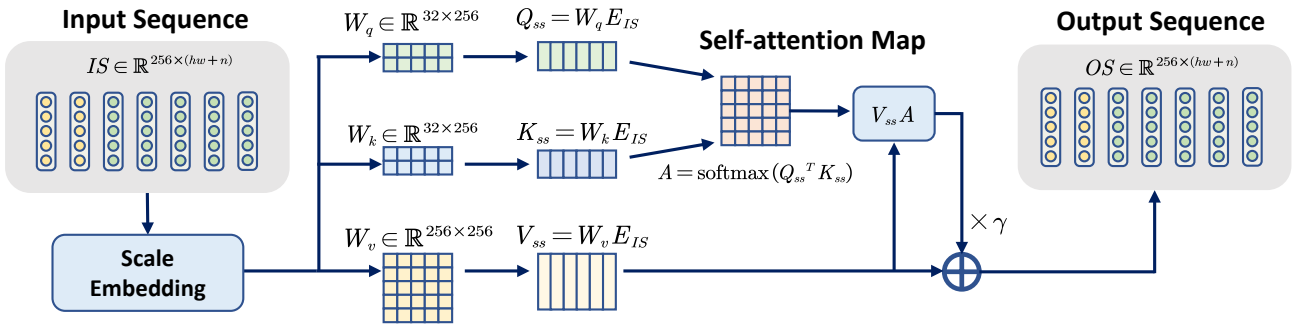

Figure 2. **Self-similarity module.** First, scale embedding is injected into the input sequence. Then each feature in the sequence aggregates the information from the rest features via self-attention and is added back to refresh the original feature.

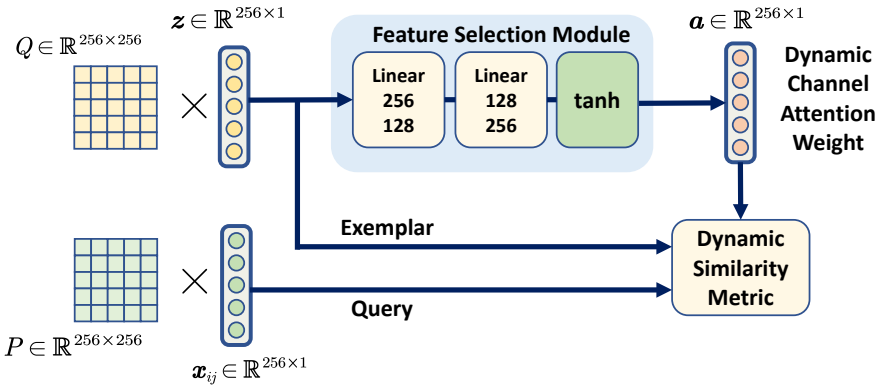

Figure 3. **Dynamic similarity module.** We add a feature selection over transformed exemplar feature so that the original bilinear similarity metric is dynamically adjusted according to exemplar-specific information.

learnable ratio γ to obtain a refreshed sequence. γ is initialized by zero so that the self-attention module will not affect the original features (*i.e.*, V_{ss}) in the early training stage. Finally, we re-split the result sequence into exemplar features and query ones, whose size is the same as that of the input sequence.

2.3. Similarity Metric

Bilinear Similarity Metric. Our bilinear similarity metric contains two transform matrices P and Q with their corresponding bias, implemented by two linear fully connected layers yielding 256 output channels.

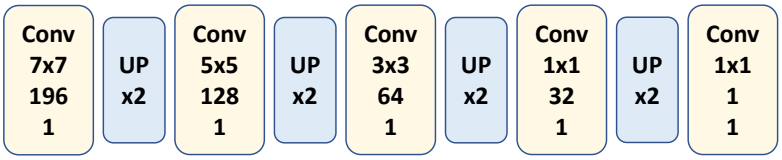

Figure 4. **Counter in BMNet and BMNet+.** UP$\times$2 denotes bilinear upsampling operation with a scale factor of 2.

Similarity Metric	Val MAE	Val MSE	Test MAE	Test MSE
BM-O	19.09	66.19	16.51	89.65
BM-64	19.99	69.00	16.80	86.94
BM-128	17.60	63.15	15.95	98.28
BM-256 (default)	19.06	67.95	16.71	103.31

Table 1. The comparison between the original bilinear similarity metric BM-O and our modifications BM-k where $k = 64, 128, 256$.

Dynamic Similarity Metric. The dynamic similarity metric adds a feature selection module over $Q\boldsymbol{z} + b_{\boldsymbol{z}}$ (the exemplar feature transformed with Q). It consists of a three-layer perceptron (Linear-ReLU-Linear) as detailed in Fig. 3. Additionally, we add a `tanh` activation after the feature selection module and add a bias of 1 to the output weights so that the weights are limited within the range $[0, 2]$.

2.4. Counter

As shown in Fig. 4, the counter consists of layers of convolutions and bilinear upsampling. With this module, the predicted density map is recovered to the same size as that of query image, which is the common choice in previous methods [3, 4].

3. The Potential of Decomposing Original Bilinear Similarity

Given two vectors $\boldsymbol{x}_{ij} \in \mathbb{R}^{d\times 1}$ and $\boldsymbol{z} \in \mathbb{R}^{d\times 1}$, the bilinear similarity metric derived from the bilinear model [2] works in the form $\boldsymbol{x}_{ij}^T W \boldsymbol{z}$, where $W \in \mathbb{R}^{d\times d}$ is a learnable matrix. In our BMNet, we decompose it into two transform matrices $P \in \mathbb{R}^{d\times k}$ and $Q \in \mathbb{R}^{d\times k}$ for exemplar and query image. Here we test BMNet with different settings: the originally not decomposed (BM-O), $k = 64$ (BM-64), $k = 128$ (BM-128), and $k = 256$ (BM-256). The results are reported in Table 1. First compare our decomposed BM-k with different settings of k. It can be observed that BM-128 achieves the best performance, which shows the potential of our method with careful parameter tuning. Then compare our decomposed BM-k (especially BM-128) with BM-O. It can be observed that the decomposition operation brings benefit. We suppose it might indicate that learning separate transformation matrices for exemplar and query image may produce more fine-grained metrics and hence present better results. Here we choose $k = d = 256$ as our default setting since more dimensions may contain more patterns to facilitate the following dynamic feature selection module. Careful parameter tuning is not our focus in this work.

4. Visualizations of Dynamic Channel Attention Weights

More visualizations of dynamic channel attention weights are shown in Fig. 5. We visualize the mean channel attention weights of some visually dissimilar categories and similar categories. It can be observed that for dissimilar categories, channel attention weights exhibit obvious differences on specific channels, while they show more consistency for visually similar categories. This justifies that dynamic similarity metric learns to focus on discriminative patterns for each exemplar.

5. Visualizations of Predicted Density Maps and Counting Results

We add more visualizations of predicted results in Fig. 7 and Fig. 8 to show the advantage of our proposed BMNet and BMNet+ over the previous state-of-the-art FamNet+ [3].

6. The Benefit of Adding Query Features for Counting Regression

In the illustration of main pipeline (cf. Fig. 2 in the main paper), the similarity map is concatenated with query features before fed into count regressor. And in Sec. 4.4 of the main paper, we quantatively show the benefit of adding query features.

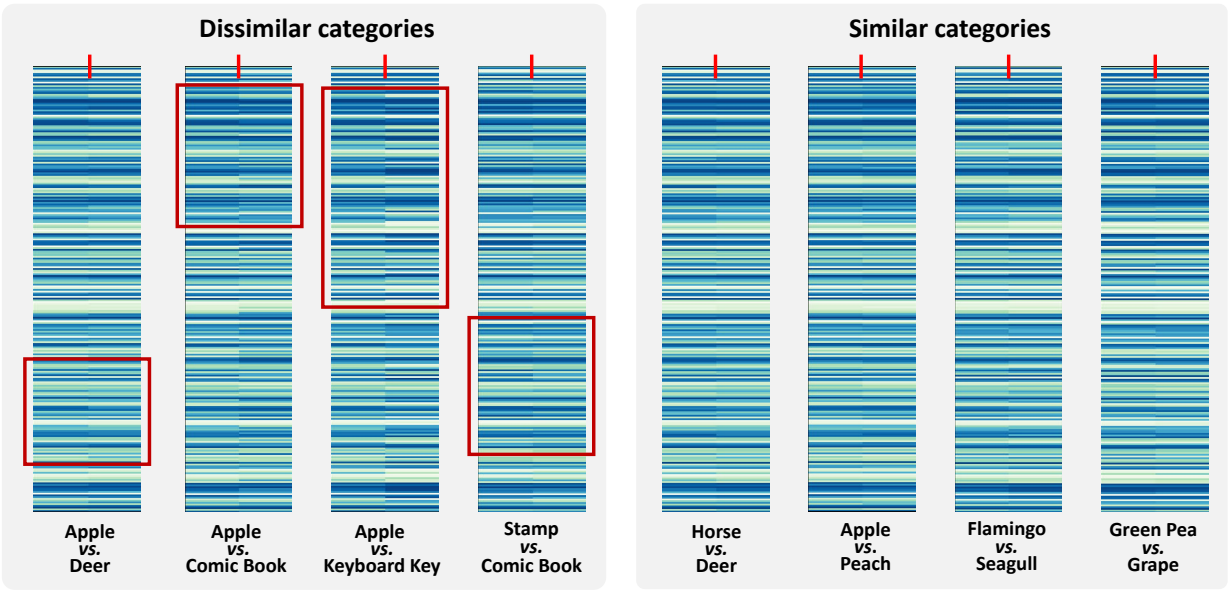

Figure 5. Comparison of mean dynamic channel attention weights for visually dissimilar categories and similar categories. In each column case, the short red line splits the channel attention weights for two different categories. Inconsistent attention weights for the two categories exhibit fault lines, which is accentuated between dissimilar categories (the red boxes).

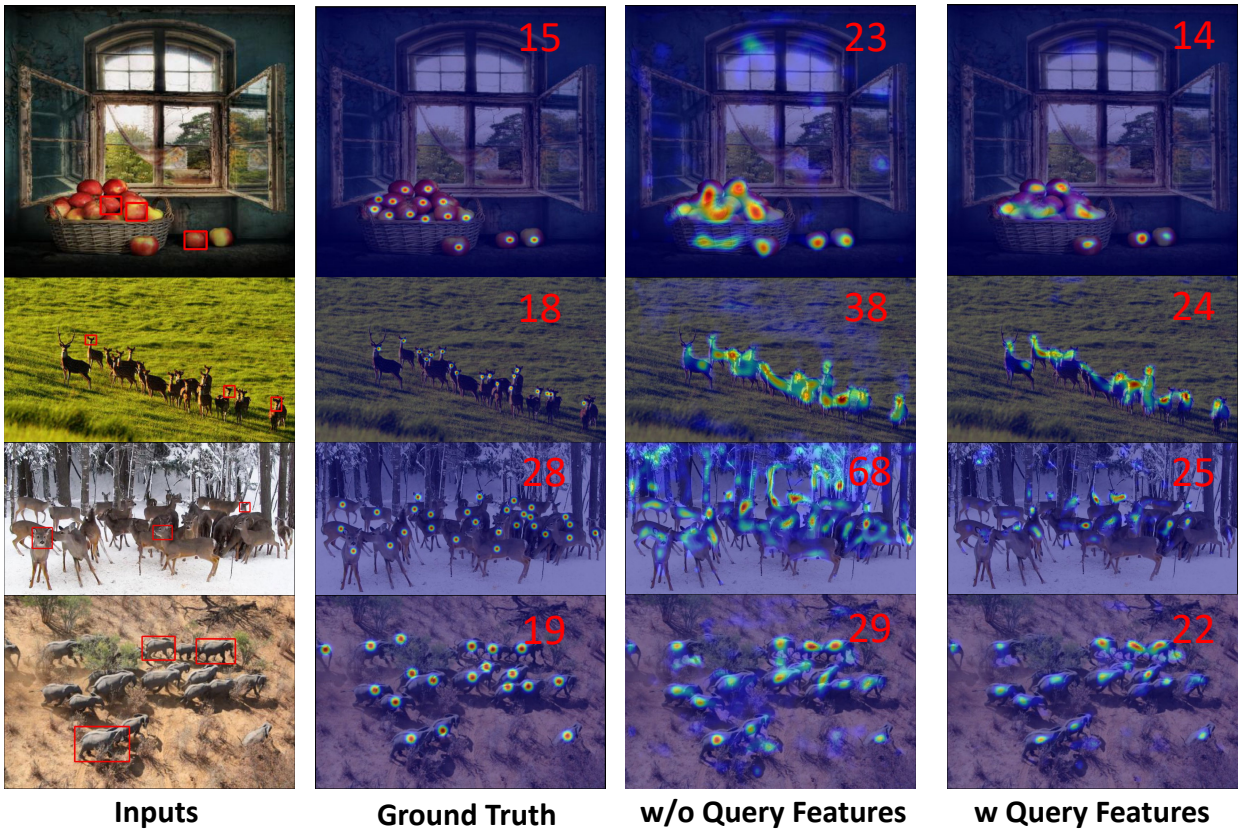

Figure 6. Visualizations for BMNet with and without adding query features before the count regressor.

Number of Exemplars	Self-similarity	Val MAE	Val MSE	Test MAE	Test MSE
1	Without	19.92	71.58	17.08	96.71
1	With	17.89	61.12	16.89	96.65
2	Without	18.86	69.63	15.97	101.34
2	With	16.03	58.65	16.16	97.18
3	Without	17.95	68.02	15.46	100.36
3	With	15.74	58.53	14.62	91.83

Table 2. The benefit of self-similarity module under few-shot scenarios.

Here we additionally present some qualitative results in Fig. 6 to help understand the benefit. As can be observed, when only using similarity map for count regression, the model tends to predict false positive response on background textures, *e.g.*, wall in the first row or grass in the second row. However, with the semantic information contained within the query features, the model can better distinguish the background textures and instances, and correct the above mistakes.

7. The Benefit of Self-similarity Under Few-shot Scenarios

Class-agnostic counting in the current benchmark is a few-shot task since the number of exemplars within each test query image is limited to 3 in FSC147 dataset. In this few-shot scenario, intra-class variation imposes challenge on the algorithms as stated in the main paper. The aim of this section is to show that, our Self-Similarity Module with scale embedding (SSM) can follow its design to address this challenge.

Specifically, we test our method BMNet+ with or without SSM when the number of exemplars n is limited, *i.e.*, $n = 1, 2, 3$. The results are reported in Table 2. It can be observed that, SSM offers help in all the cases, which justifies our proposition. It is also worth mentioning that, executing SSM when $n = n_1$ can achieve comparable (cf. test MAE) even better (cf. val MAE) performance than removing SSM when $n > n_1$. *I.e.*, in the test scenarios, our SSM brings more robustness towards the intra-class variation than directly increasing exemplar numbers.

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, pages 770–778, 2016. 1

[2] Hamed Pirsiavash, Deva Ramanan, and Charless Fowlkes. Bilinear classifiers for visual recognition. In *Proc. Adv. Neural Inf. Process. Syst.*, volume 22, 2009. 3

[3] Viresh Ranjan, Udbhav Sharma, Thu Nguyen, and Minh Hoai. Learning to count everything. In *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, pages 3393–3402, 2021. 1, 2, 3

[4] Shuo-Diao Yang, Hung-Ting Su, Winston H. Hsu, and Wen-Chin Chen. Class-agnostic few-shot object counting. In *Proc. Winter Conf. Appl. Comput. Vis.*, pages 869–877, 2021. 3

[5] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arxiv1805.08318*, 2019. 1

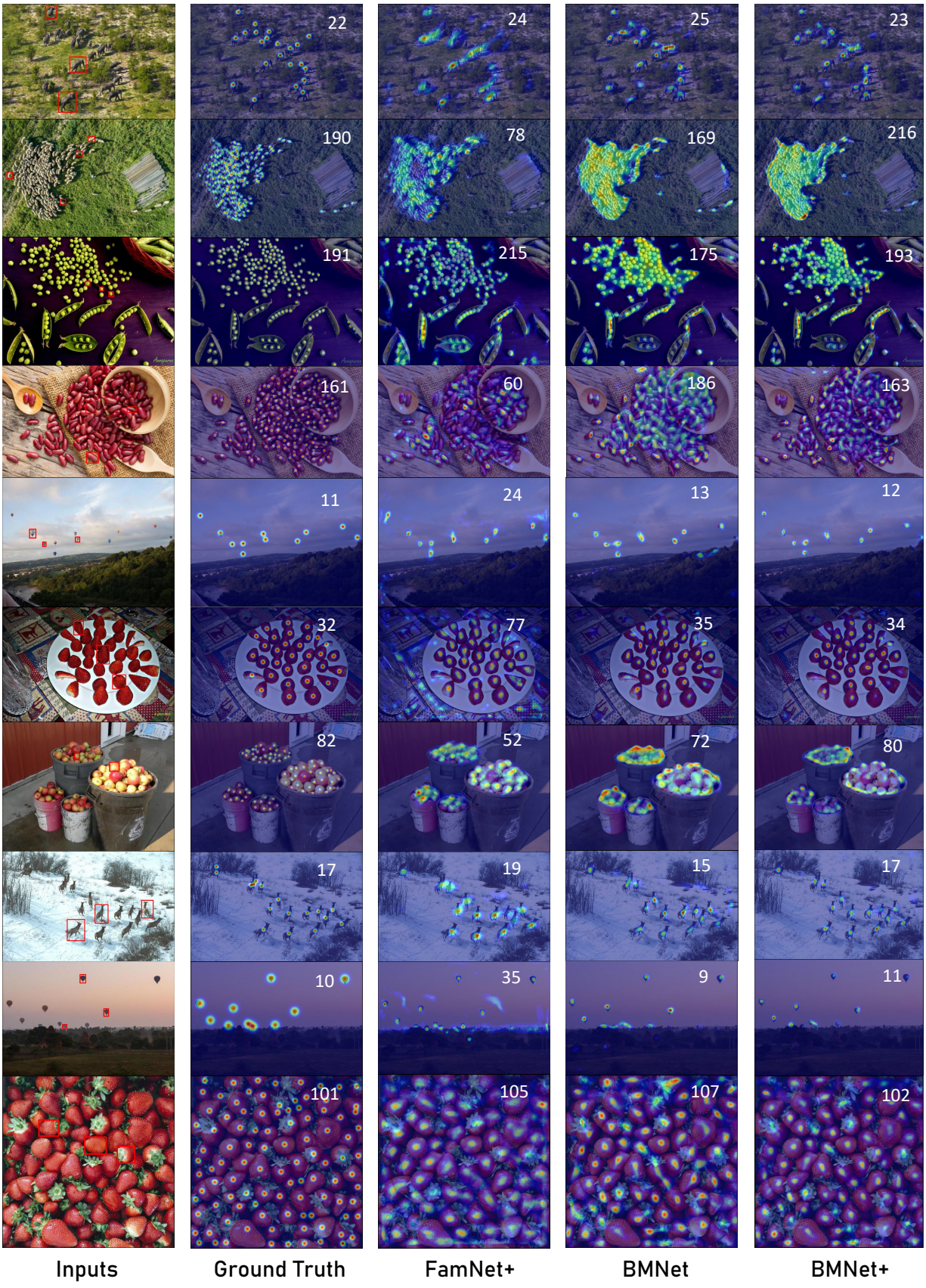

Figure 7. More visualizations of density maps and counting results predicted by different methods.

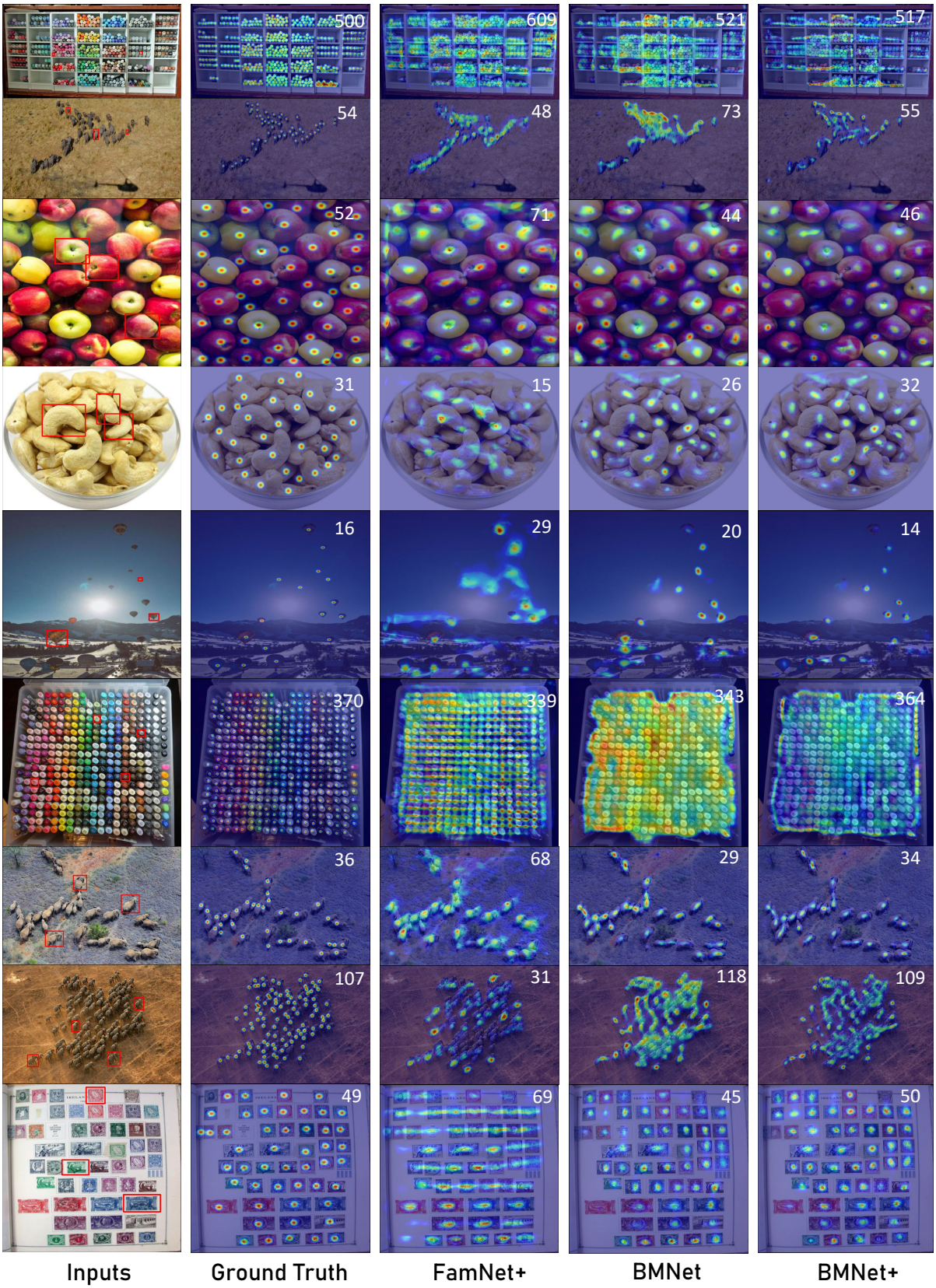

Figure 8. More visualizations of density maps and counting results predicted by by different methods.