Supplementary Material for Scale-Prior Deformable Convolution for Exemplar-Guided Class-Agnostic Counting

Wei Lin[1]
wlin38-c@my.cityu.edu.hk
Kunlin Yang[2]
yangkunlin@sensetime.com
Xinzhu Ma[3]
xinzhu.ma@sydney.edu.au
Junyu Gao[4]
gjy3035@gmail.com
Lingbo Liu[5]
liulingbo918@gmail.com
Shinan Liu[2]
liushinan@sensetime.com
Jun Hou[2]
houjun@sensetime.com
Shuai Yi[2]
yishuai@sensetime.com
Antoni B. Chan[1]
abchan@cityu.edu.hk

[1] Department of Computer Science
City University of Hong Kong
Kowloon, Hong Kong SAR
[2] SenseTime Group Limited
Beijing, P.R. China
[3] The University of Sydney
NSW 2006, Australia
[4] School of Artificial Intelligence, Optics and Electronics
Northwestern Polytechnical University
Xi'an, P.R. China
[5] The Hong Kong Polytechnic University
Hunghom, Hong Kong SAR

1 Overview

This supplementary material is organized as follows:

- In Section 2, more details about our SPDCN are presented, e.g., architecture of $\mathcal{C}$, $\mathcal{G}$ and $\mathcal{R}$, and the stucture of density decoder, and the reason why Vgg-19 is adopted as the architecture of backbone.
- In Section 3, we analyze how different number of exemplars affects counting results in exemplar-guided class-agnostic counting.
- In Section 4, we present results on a subset of FSC-147 only containing categories in COCO [2] to compare our model with detection-based methods;
- In Section 5, we discuss how these hyperparameters α, β, μ and σ in scale-sensitive generalized loss affect the performance of SPDCN.

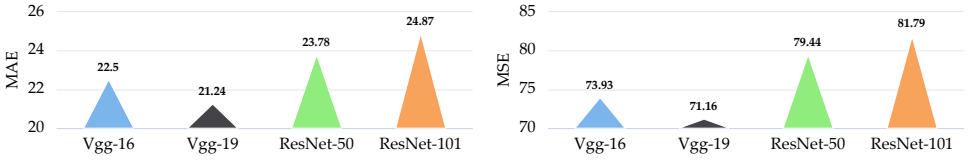
© 2022. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

non-linear $\mathcal{C}$	non-linear $\mathcal{G}$
Input size: $n \times h \times w$	*Input size:* 2
Conv_3x3$(n, 64)$ ReLU Conv_3x3$(64, 32)$	Linear$(2, 64)$ ReLU Linear$(64, 32)$
Output d_c*:* $32 \times h \times w$	*expand to* d_g*:* $32 \times h \times w$
non-linear $\mathcal{R}$	
Concatenate $\mathcal{C}$ *and* $\mathcal{G}$*:* $64 \times h \times w$	
Conv_3x3$(64, 32)$ ReLU Conv_3x3$(32, 18)$	
Output size: $18 \times h \times w$	

Table 1: Details about $\mathcal{C}$, $\mathcal{G}$ and $\mathcal{R}$.

layer		output feature size
input size		$512 \times \frac{1}{8}h \times \frac{1}{8}w$
Conv	3x3	$256 \times \frac{1}{8}h \times \frac{1}{8}w$
ReLU		
Conv	3x3	$256 \times \frac{1}{8}h \times \frac{1}{8}w$
ReLU		
Pixel Shuffling		$4 \times h \times w$
Conv	3x3	$1 \times h \times w$
ReLU		

Table 2: The structure of decoder.



(a) MAEs of different vanilla backbones. (b) MSEs of different vanilla backbones.

Figure 1: MAEs/MSEs of different vanilla backbones.

- In Section 6, more representative visual results are displayed to show the advantage of our proposed SPDCN.

2 Network Details

The offsets in the scale-prior deformable convolution are generated through three non-linear functions $\mathcal{C}$, $\mathcal{G}$, and $\mathcal{R}$. As shown in Table 1, $\mathcal{C}$ is the function for local embedding d_c, following a typical non-linear structure *Conv-ReLu-Conv*; $\mathcal{G}$ is similar to $\mathcal{C}$, but *Linear* function is adopted since the input only contains two elements ($\bar{h}$ and $\bar{w}$); $\mathcal{R}$ also has the architecture of *Conv-ReLU-Conv*, which is designed to firstly concatenate the local embedding d_c and global embedding d_g, and then generate offsets for deformable convolution. The output has 18 channels, representing the offsets aligning two axes for a 3×3 convolutional kernel.

With the output feature maps from the scale-prior deformable backbone, we segment out regions containing counted objects by cosine similarity. After segmentation, the size of density feature maps is $512 \times \frac{1}{8}h \times \frac{1}{8}w$, in which h and w are the height and width of the input image. To construct the final density map $D \in \mathbb{R}^{1 \times h \times w}$, the feature should be firstly compressed and interpolated. In our model, compression is done by two convolution layers, and the upsampling method is a pixel shuffling operator [7]. The structure of the decoder is displayed in Table 2.

Another question is why Vgg-19 [8] is adopted as the backbone architecture. In Figure 1, we present empirical results while various architectures are used in our models. Results show that Vgg-19 achieves the lowest estimations errors. Notably, the scale-prior deformable convolution is not inserted in these backbones. We believe that the best vanilla feature extractor can also perform better than others when the scale-prior deformable convolution is inserted.

Exemplar number	# 1		# 2		# 3	
Method	MAE	MSE	MAE	MSE	MAE	MSE
FamNet [4]	26.55	77.01	24.09	72.37	23.75	69.07
BMNet [6]	17.89	61.12	16.03	58.65	15.74	58.53
SPDCN(ours)	**16.36**	**53.94**	**15.64**	**51.17**	**14.59**	**49.97**

Table 3: Comparison with other methods when different number of exemplars are used.

Method	Validation COCO		Test COCO	
	MAE	MSE	MAE	MSE
Faster R-CNN [5]	52.79	172.46	36.20	79.59
RetinaNet [3]	63.57	174.36	52.67	85.86
Mask RCNN [1]	52.51	172.21	35.56	80.00
FamNet [4]	39.82	108.13	22.76	45.92
BMNet [6]	**26.55**	93.63	12.38	24.76
SPDCN(ours)	29.11	**93.12**	**11.87**	**24.50**

Table 4: Comparison to pre-trained object detectors on the COCO subset of FSC-147

3 Analysis on Exemplar Number

The number of exemplars is a significant part of class-agnostic counting, so here we present experimental results of models trained with different numbers of exemplars.

As displayed in Table 3, estimation errors decrease gradually with the increase of the number of exemplars. Our SPDCN can get the lowest MAE of 16.36 when given only one exemplar, which is even better than FamNet [4] with three exemplars. When giving two exemplars, SPDCN's MAE(MSE) is reduced to 15.64(51.17). For comparison, BMNet achieves an MAE(MSE) of 16.03(58.65) under this setting. When three exemplars rae provided, our model can utilize the new information well and obtains MAE of 14.59.

4 COCO Subset

In this section, we follow the setting of FamNet [4] and compare our model with detection-based methods. Since thousands of samples are required to train a detector, a subset of FSC-147 [4] only containing categories existing in COCO [7] is used to evaluate counting models. As shown in Table 4, detection-based counters like Faster R-CNN [5], RetinaNet [3] and Mask R-CNN [1] perform worse on counting tasks and all of them have MAEs larger than 50 on the validation set, while density-estimating methods work better. FamNet reduces MAE on validation-COCO and test-COCO by about 10 and 15, respectively; BMNet [6] has MAE and MSE of 26.55 and 12.38. However, our SPDCN's MAE on validation COCO is larger than BMNet, but our MSE is smaller. Besides, SPDCN achieves the lowest estimation errors on the test-COCO set (MAE: 11.87, MSE: 24.50).

5 Hyperparameters in Scale-Sensitive Generalized Loss

Scale factors in scale-sensitive generalized loss are defined by a modified sigmoid function:

$$\mathcal{S}(k) = \frac{\alpha}{1+\exp[-(k-\mu)/\sigma]} + \beta, \tag{1}$$

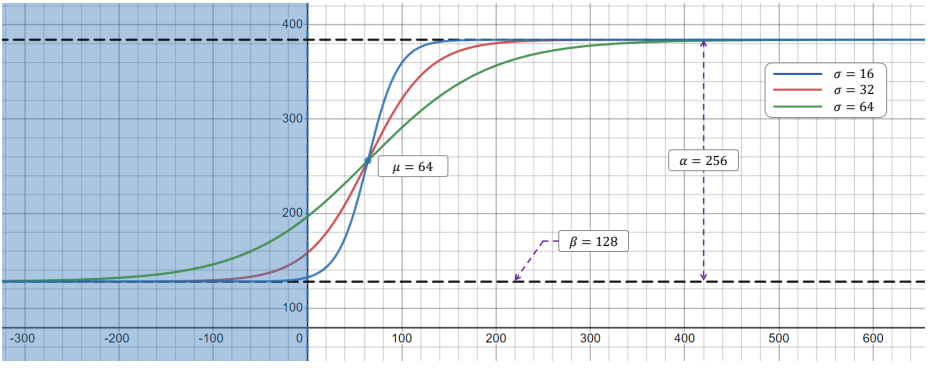


Figure 2: Visualization of the modified sigmoid function. The shaded region will never be used since the input k (i.e., $\bar{h}$ or $\bar{w}$) is always positive.

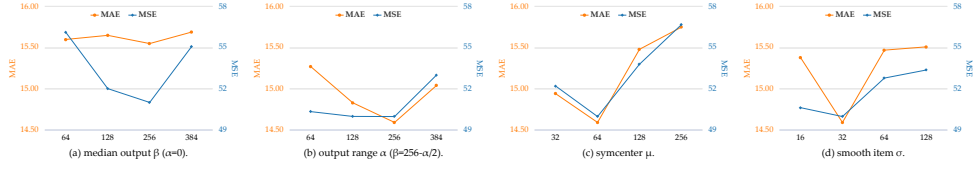


Figure 3: Ablation study on hyperparameters in the scale-sensitive generalized loss.

so that we have $s_h = \mathcal{S}(\bar{h})$ and $s_w = \mathcal{S}(\bar{w})$. There are four hyperparameters: μ, σ, α and β. Figure 2 visualizes three examples with different σ. As shown in the figure, μ defines the symcenter of the S-curve; σ is used to smooth the curve; α and β control the range and minimum value of output. Here we conduct experiments by applying SPDCN to scale-sensitive generalized loss in different settings, demonstrating how the loss function helps exemplar-guided class-agnostic counting.

Firstly, we set α as 0 to observe the effect of β, in which case the loss function degenerates into a vanilla generalized loss instead of the scale-sensitive one. As displayed in Figure 3(a), MAE does not change dramatically, but MSE decreases by about 5 when β changes from 64 to 256. Notably, β represents the upper and lower bound simultaneously since $\alpha = 0$ here.

Next, we discuss the effect of α. Here we fix μ and σ as 64 and 32. For β, we set $\beta = 256 - \frac{\alpha}{2}$ to keep the median (instead of the lower bound) of output value equal to 256, so that the range of output is $[256 - \frac{\alpha}{2}, 256 + \frac{\alpha}{2}]$. Figure 3(b) demonstrate how MAE and MSE change with the increase of α, and it shows that the lowest MAE/MSE is obtained when $\alpha = 256$ and $\beta = 128$.

By fixing α and β, we explore the influence of μ and σ. In Figure 3(c), σ is set as 32, and μ changes from 32 to 256. In Figure 3(d), σ gradually increase to 128 as $\mu = 64$. Both figures show that neither a large or small μ/σ is harmful to counting performance. The lowest estimation errors are given when μ and σ are set to 64 and 32, respectively. The relationship between scale factor $s_w(s_h)$ and average length of side $\bar{w}(\bar{h})$ is displayed by the red line in Figure 2.

According to these ablation studies, we set α, β, μ and σ to 256, 128, 64 and 32, respectively.

6 More visualizations

Fig. 4 and Fig. 5 present more visualization results, in which we compare our SPDCN with FamNet [4] and BMNet [6]. The corresponding segmentation masks computed by SPDCN are also presented. These examples show that SPDCN performs better than other models, and the prediction of our model is more accurate.

References

[1] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[2] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[3] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[4] Viresh Ranjan, Udbhav Sharma, Thu Nguyen, and Minh Hoai. Learning to count everything. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3394–3403, 2021.

[5] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015.

[6] 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/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9529–9538, June 2022.

[7] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016.

[8] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

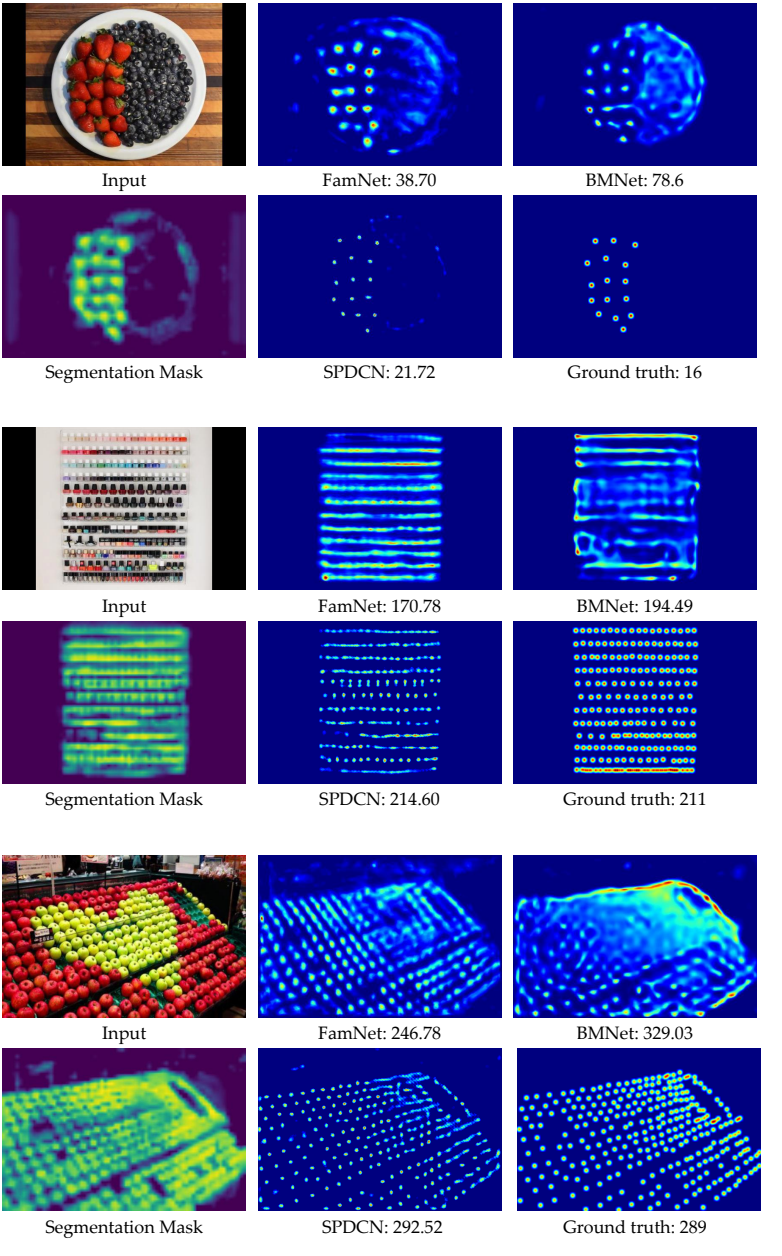


Figure 4: More visualizations of density maps predicted by different methods. Segmentation masks produced by SPDCN are also presented.

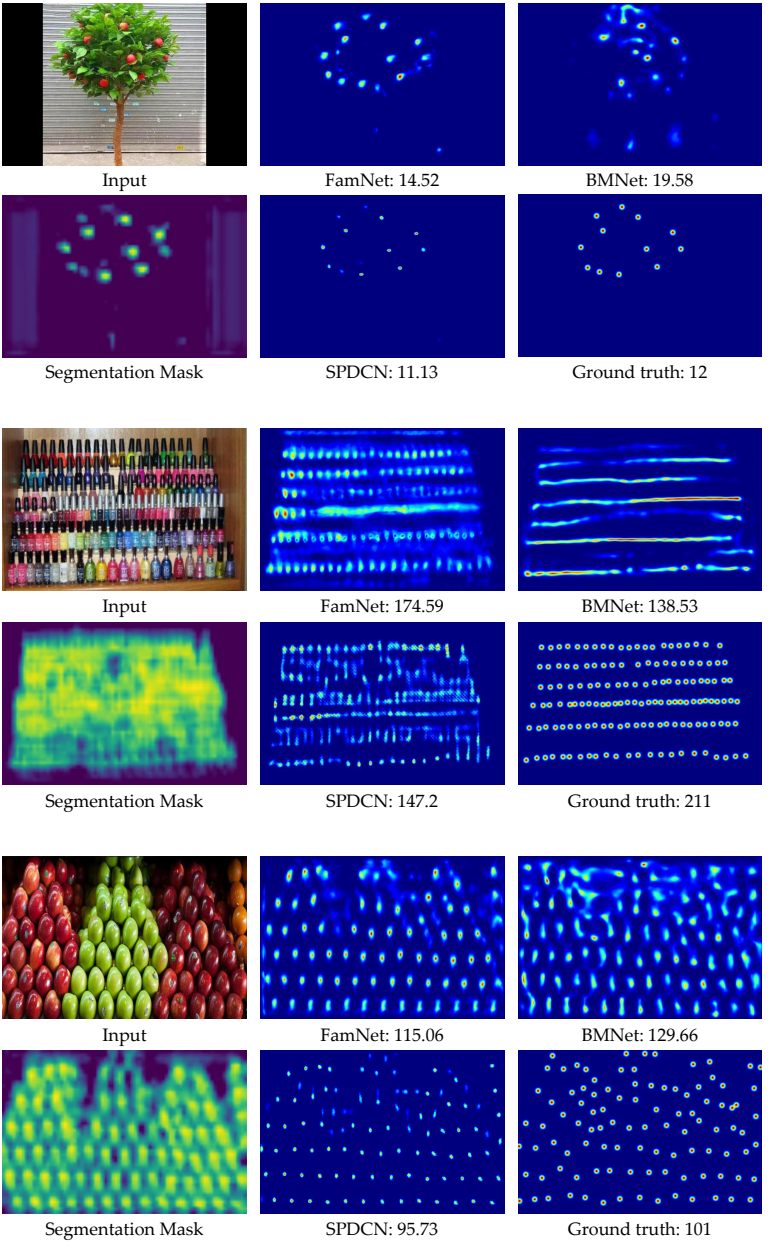


Figure 5: More visualizations of density maps predicted by different methods. Segmentation masks produced by SPDCN are also presented.