RETHINKING MULTI-SCALE IN CROWD COUNTING: A SINGLE-COLUMN NEURAL NETWORK WITH LAYER AGGREGATION

Mengyuan Xi, Shengnan Ma, Hua Yan[*]

College of Electronics and Information Engineering, Sichuan University, Chengdu Sichuan, China

ABSTRACT

The accuracy of crowd counting is significantly subject to the huge scale variation. Many existing methods focus on utilizing multi-column or multi-branch structures to deal with this problem. However, these approaches have limited ability in capturing multi-scale features because each sub-network path of multi-column framework attempts to minimize the regression loss function independently and to predict the correct density map for images with all head scales. The features learned by different column are totally independent and this manner seriously limits the learning of effective features. Besides, this architecture has larger parameters and computational burdens compared to single-column network with same depth. In this paper, we propose a single-column network (SLANet) with layer aggregation mechanism (LAM), which is composed of aggregation across layers and channels and only aggregates layers across a network to obtain diverse scale information for crowd counting. In addition, we introduce a novel weighted normalized loss which alleviates the effect of noise background and uneven crowd distribution. Our SLANet outperforms state-of-the-art crowd counting methods according to extensive experiments carried out on five mainstream datasets.

Index Terms— Crowd counting, feature fusion, layer aggregation, multi-scale

1. INTRODUCTION

Crowd counting remains challenging due to great scale changes in different crowd scenes. As shown in Fig. 1, there exists a dramatic scale variation within a single image. What's more, the size of people head has a huge diversity between other images and different datasets. Therefore, the key to improve the accuracy of crowd counting method is how to better capture multi-scale features.

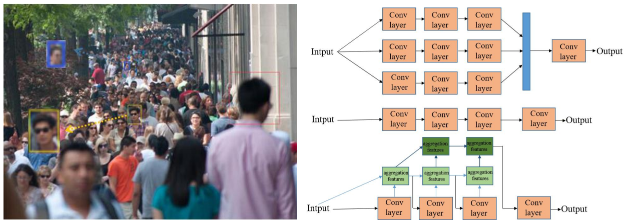

Fig. 1. A sample for illustrating the huge scale variation problem and review of related solution method. The top, middle and bottom of the right part show the architecture of multi-colum, regular single-clomun and single-column with our proposed layer aggregation mechanism network respectively. The first structure is early introduced in crowd counting (MCNN [1], 2016; Switch-CNN [6], 2017; CP-CNN [7], 2017). The most famous of the second structure is called CSRNet [8] (2018) which utilize dilated convolution to deliver larger recptive fields. However, it does not consider obtaining multi-scale information. After that, many multi-column networks were presented, including MFANet [2] which suffer from inherent drawbachks like we discussed before. The simple illumination of our proposed method is listed in the bottom. We rethink the acquisition mechanism of multi-scale features in crowd counting which can learn effective multi-scale features while having same parameters compared to regular single-column network. Based on the proposed layer aggregation mechanism, we can explore deeper network (more convolutions) and more aggregations (deeper recursion depth). Moreover, it can boost exsiting regular sing-column methods like CSRNet to a better performance.

Many researchers have made tremendous efforts to deal with this issue by either using multi-column convolutional neural networks or multi-branch [1]-[5] structure with different receptive fields to extract multi-scale features for generating quality density maps. Although these techniques are effective in alleviating the problems, most of these methods suffer inherent algorithm drawbacks. In particular, each sub-network path of this framework attempts to minimize the regression loss function independently and to predict the correct density map for images with all head scales. The features learned by different column are totally independent and this manner seriously limits the learning of effective features. Moreover, each sub-network with specific scale can only work well on its corresponding scale and its performance on other scales is poor. Multi-column networks generally fuse features of different columns through concatenation operation. However, there is a large semantic gap between information of several columns; it is easy to cause low quality and blur of density map. The scale diversity of multi-branch methods

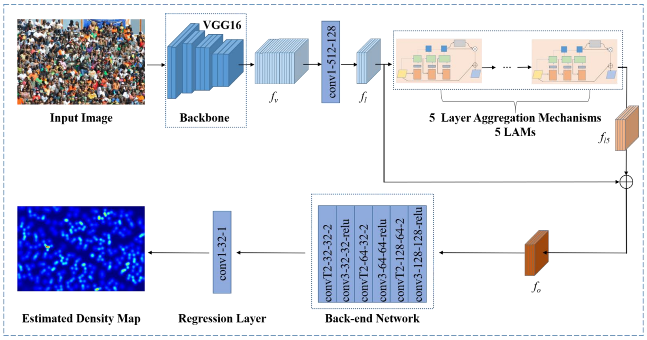

Fig. 2. Pipeline of the proposed SLANet. Convolutions are denoted as kernel size-input channels-output channels-relu while the last character of deconvolutions represents stride.

is completely restricted by the number of branches while the number of branches is not the more the better. Excessive branches can dramatically increase the complexity of the model and the risk of overfitting; it may lead to performance degradation.

Li et al. [8] introduces a single-column network that utilizes dilated convolution to deliver larger reception fields. However, they do not pay attention to capture multi-scale information and obviously ignore the rich representation hidden in the layers of network. Taking all above drawbacks into consideration, we propose a simple-structured method called SLANet, which utilizes a single-column CNN to fully extract information from different layers of the network via aggregating hierarchical features progressively. Specifically, it mainly consists of three components: a backbone network as the low-level features extractor, five LAMs to fully aggregate features of different layers, and a back-end network, composed of seven convolutional layers, for predicting density maps, as shown in Fig. 2. Taking the advantage of layer aggregation mechanism and dilated convolution, we can obtain abundant information of crowd features including multi-scale features without over-assigning parameters since we employ layer aggregation which is mainly concatenation operation (having no parameters to be trained and extra computations) instead of complicated-structured network composed of multi-branch convolutions.

The major contributions of this work are as follows: (1) A novel single-column network (SLANet) is proposed for crowd counting, which can improve the performance of crowd counting via aggregating features of different receptive fields across layers of different depth without complex feature extraction mechanism. (2) We present a layer aggregation mechanism (LAM) to extract features and further fuse rich semantic and spatial information for recognition and localization to generate quality density maps. (3) We carry out extensive experiments on five mainstream crowd counting datasets and experimental results demonstrate that our method achieves superior performance compared to other state-of-the-art methods. Detailed ablation studies verify the impact of each component in our network.

2. METHOD

Even with the depth of features in a convolutional neural network, utilizing feature maps of a separate layer to generate density maps is not enough (e.g., single-column network). The fundamental idea of our proposed SLANet model is to aggregate and reuse inherent hierarchical features of a single-column CNN for capturing multi-scale information. This idea can lower the counting error and generate high-quality density map without brutally expanding complexity of network.

SLANet Architecture: the pipeline of the proposed simple baseline framework is shown in Fig. 2. We adopt ten first convolutional layers of VGG-16 ad our backbone. After the processing of backbone network (the first ten convolutions), the size of output feature maps is 1/8 of the original input size. Then, in order to prevent widening the network, we compress the channel via 1×1 convolution which can reduce the number of parameters. After that, we sequentially stack 5 LAMs for exploiting multi-scale features. Finally, we adopt the back-end network to generate density maps by utilizing feature maps processed by front stacked LAMs. Specifically, three convolutional layers and three deconvolutional layers are utilized to make sure restore the output density maps to the original resolution. In particular, we progressively utilize deconvolution with a 2×2 kernel size and a stride of 2 to alleviate uneven overlap that deconvolution can easily have when the kernel size of deconvolution (i.e., size of output window) cannot be divided by the stride.

Layer Aggregation Mechanism: LAM is the core part of the proposed SLANet, and it is mainly composed of three dilated convolutions, a channel attention module. Based on this foundation, a series of progressive aggregations and iterative fusions are designed to further explore the considerable useful information hidden in the original network. Details of LAM is illustrated in Fig. 3.

Aggregation across layers (AAL): Multiple convolutions of small kernel have same receptive field compared to fewer convolutions of big kernel, while small kernel size can achieve higher efficiency. Consequently, 3×3 convolutions are adopted in the layer aggregation mechanism. The dilated convolution can enlarge receptive fields and does not bring the increase of parameters, which allows flexibly aggregating multi-scale contextual information. Hence, we utilize dilated convolution to replace normal convolution for delivering larger receptive fields in our method while having no extra parameters and computations. As shown in Fig. 3, we utilize three dilated convolutions with dilation rates d in {1, 2, 3} (motivated by the previous work [9]). Compared with the regular stack [8] of dilated convolution block with dilation rates in {2, 2, 2}, our block can effectively avoid the gridding effect which will results in loss of pixels and damage of information continuity. Aggregation across layers is defined as the combination of different layers throughout a network. the input features first pass through a dilated convolution and a nonlinear operation is performed. Then, we aggregate the output and input feature via concatenation. Above process is a complete aggregation operation and aggregation block is formed with involved layers. Rather than only aggregating intermediate features, we instead feed the output of an aggregation block back into the network as the input of the next aggregation block, as shown in Fig. 3. This propagates the aggregation features of all previous blocks instead of the preceding block alone to better preserve information. The progressive aggregation across layers function *P* for a series of layers d_1, d_2, d_3 with increasingly semantic and deeper information is formulated as:

$$P(d_1, d_2, d_3): \begin{cases} b_1 = cat(\sigma(c_1(f_l)), f_l) \\ b_2 = cat(\sigma(c_2(b_1)), b_1) \\ b_3 = cat(\sigma(c_3(b_2)), b_2) \end{cases} \quad (1)$$

where c_1, c_2, c_3 represent dilated convolutions with coprime dilation rates in {1, 2, 3}. σ denotes a *ReLU* function and *cat* denotes the concatenative operation. b_1, b_2, b_3 are the output of corresponding aggregation block and they will be iteratively fused by subsequent fusion nodes.

As shown in Fig. 3, after progressively aggregating different layers, fusion node is introduced to conduct iterative reuse of those aggregation features. Although a fusion node can be based on any block or layer, for simplicity and efficiency we choose a single convolution followed by a nonlinearity. This avoids overhead for fusion structures. All the nodes use 1×1 convolution to convert different number of input channels to 128 which is the same as the input channel of LAM. The main function of a fusion node is to combine and compress its inputs. This process is described as:

$$F_{fn}(x_1, x_2) = \sigma(conv(cat(x_1, x_2))) \quad (2)$$

where x_1, x_2 are the inputs of fusion node. *cat* represents the concatenative operation. *conv* is a 1×1 convolution which is utilized to change the number of channels. Then the fusion nodes learn to select and project significant information to maintain the same dimension at their output as the input of LAM, which can reduce the computation complexity and facilitate subsequent connection. Finally, we send these fusion features to channel attention module.

Aggregation across channels (AAC): Generally, different channel of feature maps has different contributions to the estimation of crowd density. Aggregating features across channels can adjust abstract semantic information according to the significance of each channel, which helps to conduct accurate density estimation. Inspired by the [58], we introduce a channel attention module to measure the significance of different channels within feature maps and selectively focus on meaningful parts for obtaining optimal visual features better. As shown in Fig. 3, in a LAM, the channel attention module consists of one global average pooling layer, one 1×1 convolution, and one nonlinear activation function. We first squeeze all feature maps $F_C \in R^{C\times H\times W}$ from the output of aggregation across layers into one dimension of $F_{avg} \in R^{C\times 1\times 1}$ through adaptive average pooling layer. Then a 1×1 convolution and a *ReLU* function are applied to rank the importance of different channels and generate corresponding weight through utilizing F_{avg} that contains global spatial context information. In short, the channel attention weight is computed as:

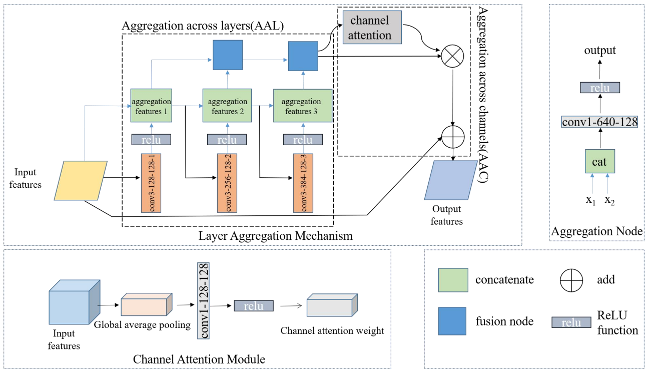

Fig. 3. The detailed architecture of layer aggregation mechanism (LAM). The convolutional layers are denoted as "conv(kernel size)-(number of filters)-(stride)".

$$C_W(F_C) = \sigma(Conv(GAP(F_C))) = \sigma(Conv(F_{avg})) \quad (3)$$

where F_C is the input feature maps and $C_W \in R^{C\times 1\times 1}$ is the output weight, *GAP* represents a global average pooling layer. Finally, the computed channel attention weights are applied to the input feature maps. These intermediate feature maps are adaptively refined through channel attention module. The overall channel attention process is summarized as:

$$F_{CO} = C_W(F_C) \otimes F_C \quad (4)$$

After the aggregation across layers and channels, we adopt a residual connection between the input and output of LAM for better propagation of features and gradients, and this result is the final output of our LAM.

Loss Function: we introduce the weighted normalized Euclidean loss as the loss function to measure the difference between the ground truth and the estimated density map generated by our model. The loss function is given as follow:

$$L_n = \frac{1}{2N} \sum_{i=1}^{N} \frac{\left\| Z(X_i;\theta) - Z_i^{GT} \right\|_2^2}{\sum_{m=1}^{W} \sum_{n=1}^{H} Z_i^{GT}(m,n)^2}$$

$$weight\_mask = Z_i^{GT} \otimes k + 1 \quad (5)$$

$$L_{wn} = L_n \otimes weight\_mask$$

where *(m, n)* and *W*×*H* represent the pixel position and size of the ground truth density map, respectively. N is the size of training batch and *Z(X*$_i$*;θ)* is the output generated by

Table 1. Comparison with state-of-the-art methods on five crowd datasets. Bold denotes the best result.

Method	Network Architecture	Part A		Part B		Mall		UCF_CC_50		UCF-QNRF		Parameters (M)
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	
MCNN [1]	Multi-column	110.2	173.2	26.4	41.3	2.24	8.5	377.6	509.1	277.0	426.0	0.13
Switch-CNN [6]	Multi-column	90.4	135.0	21.6	33.4	1.62	2.10	318.1	439.2	228	445	-
CP-CNN [7]	Multi-column	73.6	106.4	20.1	30.1	-	-	298.8	320.9	-	-	68.40
CSRNet [8]	Single-column	68.2	115.0	10.6	16.0	-	-	266.1	397.5	-	-	16.26
SANet [3]	Multi-branch	67.0	104.5	8.4	13.6	-	-	258.4	334.9	-	-	1.39
CAN [5]	Multi-branch	62.3	100.0	7.8	12.2	-	-	212.2	243.7	107.0	183.0	18.10
MFANet [2]	Multi-branch	58.5	**93.4**	7.2	11.6	1.59	2.03	221.3	279.7	97.7	166	19.45
SLANet (ours)	Single-column	**56.5**	94.5	**7.0**	**10.7**	**1.56**	**1.98**	**156.1**	**186.6**	**96.9**	**163.4**	13.27

SLANet with parameters denoted as θ. X_i is the input image while Z_i^{GT} represents the ground truth of the input image X_i. The normalized loss L_n can balance the contribution of samples with different density distribution while the weight mask aims to suppress the effect of noise background through changing the pixel value from range of 0 to 1 to range of 1 to k+1.

3. EXPERIMENTS

We crop each image at different locations into 8 patches with 1/4 size and horizontal flip for data augmentation. Our SLANet is trained in an end-to-end fashion manner. The first 10 convolution layers are initialized from a VGG-16 model pre-trained on ImageNet and the rest trainable weights are randomly initialized by a Gaussian distribution with the mean of 0 and the standard deviation of 0.01. We use RMSprop optimizer with a fixed learning rate of 0.00001 to minimize the loss function.

Experiment Results: We utilize the mean absolute error (MAE) and mean squared error (MSE) as the evaluation metric [10]-[13]. The effectiveness of the proposed method is evaluated and compared against other newest methods on five crowd counting benchmark datasets: ShanghaiTech Part A/B, UCF-QNRF, UCF_CC_50, and Mall. All results are shown in the Table 1. On Part A and Part B, we reach the best 56.5 and 7.0 MAE compared to the state-of-the-art approaches. The results on Mall dataset show our method achieve a very low counting error of 1.56 and MSE of 1.98. We achieve 29.46% lower MAE than CAN. We also obtain the superior performance against other methods on the UCF-QNRF dataset. Compared to most multi-column methods, our method also has fewer parameters. We also list some visual results in the Fig. 4.

Ablation Study: As shown in Table. 2, we gradually enrich the network with aggregation across layers and channels to verify their effectiveness. We also study how the number of LAMs affect the performance of model. As the number increases, it shows an upward trend of performance, whereas when it comes to 6, the accuracy begins to decrease. It may result from the tremendous model complexity, which make the network hard to learn. Finally, we evaluate the weighted normalized loss (Lwn) with normal Euclidean loss (Le). The results demonstrate the L_{wn} also contribute to the improvement of the counting performance.

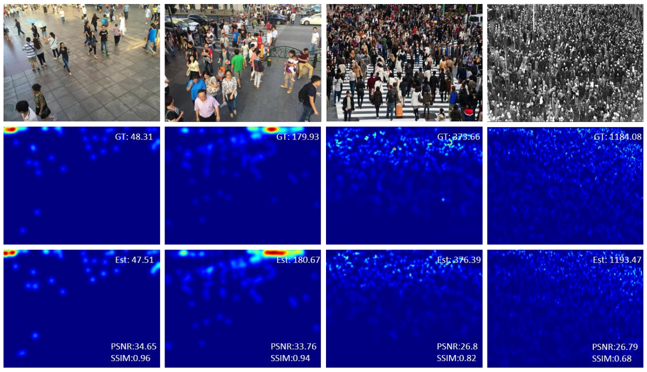

Fig. 4. Visual results on Part A/B. GT: ground truth; Est is estimated results. PSNR and SSIM are also reported.

Table 2. Ablation studies conducted on the Part A dataset.

Method	MAE	MSE
Baseline	62.8	105.6
Baseline + AAL	60.1	95.5
Baseline+AAL+AAC (SLANet)	**56.5**	**94.5**
0 LAM	64.1	102.0
1 LAM	58.3	92.8
2 LAMs	57.9	**91.7**
3 LAMs	58.1	92.5
4 LAMs	57.2	**91.7**
5 LAMs	**56.5**	94.5
6 LAMs	58.6	100.6
L_e	62.1	105.4
L_{wn}	**56.5**	**94.5**

4. CONCLUSION

In this work, we rethink the multi-scale in crowd counting and propose a single-column network to address the scale variation problem, named SLANet. Extensive experiments demonstrate that LAM enables the SLANet achieve optimal performance on five datasets with a single-column architecture. We efficiently aggregate feature hierarchy instead of being limited to designing redundant network with deeper or wider architectures brutally. In future work, we will explore the detail information of differences in multi-scale features between these two architectures and study the corresponding lightweight network design.

5. REFERENCES

[1] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma, "Single-image crowd counting via multi-column convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 589–597.

[2] F. Zhu, H. Yan, X. Chen, T. Li, and Z. Zhang, "A multi-scale and multi-level feature aggregation network for crowd counting," Neurocomputing, vol. 423, pp. 46-56, 2021.

[3] Xinkun Cao, Zhipeng Wang, Yanyun Zhao, and Fei Su, "Scale aggregation network for accurate and efficient crowd counting," in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 734–750.

[4] X. Jiang, L. Zhang, M. Xu, T. Zhang, P. Lv, B. Zhou, X. Yang, and Y. Pang, "Attention scaling for crowd counting," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2020, pp. 4706-4715.

[5] Weizhe Liu, Mathieu Salzmann, and Pascal Fua, "Context-aware crowd counting," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 5099–5108.

[6] Deepak Babu Sam, Shiv Surya, and R. Venkatesh Babu, "Switching convolutional neural network for crowd counting," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2017, pp. 4031–4039.

[7] Vishwanath A. Sindagi and Vishal M. Patel, "Generating high-quality crowd density maps using contextual pyramid cnns," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1861–1870.

[8] Yuhong Li, Xiaofan Zhang, and Deming Chen, "Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1091–1100.

[9] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," in *Proceedings of the IEEE winter conference on applications of computer vision*, 2018, pp. 1451-1460.

[10] Y. Wang, J. Hou, X. Hou, and L. P. Chau, "A self-training approach for point-supervised object detection and counting in crowds," IEEE Transactions on Image Processing, vol. 30, pp. 2876–2887, 2021.

[11] D. B. Sam, S. V. Peri, N. Mukuntha, and R. V. Babu, "Going beyond the regression paradigm with accurate dot prediction for dense crowds," in IEEE Winter Conference on Applications of Computer Vision, 2020, pp. 2853–2861.

[12] G. Gao, J. Gao, Q. Liu, Q. Wang, and Y. Wang, "Cnn-based density estimation and crowd counting: A survey," *arXiv preprint arXiv:2003.12783*, 2020.

[13] Y. Wang, J. Hou, and L. P. Chau, "Object counting in video surveillance using multi-scale density map regression," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 2422–2426.