DLMP-Net: A Dynamic Yet Lightweight Multi-pyramid Network for Crowd Density Estimation

Qi Chen[1,2], Tao Lei[1,2(✉)], Xinzhe Geng[1,2], Hulin Liu[1,2], Yangyi Gao[1,2], Weiqiang Zhao[3], and Asoke Nandi[4]

[1] Shaanxi Joint Laboratory of Artificial Intelligence, Shannxi University of Science and Technology, Xi'an 710021, China
leitao@sust.edu.cn
[2] School of Electronic Information and Artificial Intelligence, Shannxi University of Science and Technology, Xi'an 710021, China
[3] CETC Northwest Group Co., Ltd., Xi'an 710065, China
[4] Brunel University London, Uxbeidge, Middlesex UB8 3PH, UK

Abstract. The current deep neural networks used for crowd density estimation face two main problems. First, due to different surveillance distance from the camera, densely populated regions are characterized by dramatic scale change, thus using vanilla convolution kernels for feature extraction will inevitably miss discriminative information and reduce the accuracy of crowd density estimation results. Second, popular networks for crowd density estimation still depend on complex encoders with a large number of parameters, and adopt fixed convolutional kernels to extract image features at different spatial positions, resulting in spatial-invariance and computation-heavy. To remedy the above problems, in this paper, we propose a Dynamic yet Lightweight Multi-Pyramid Network (DLMP-Net) for crowd density estimation. The proposed DLMP-Net mainly makes two contributions. First, we design a shuffle-pyramid feature extraction and fusion module (SPFFM), which employs multi-dilated convolution to extract and fuse various scale features. In addition, we add group and channel shuffle operation to reduce the model complexity and improve the efficiency of feature fusion. Second, we introduce a Dynamic Bottleneck Block (DBB), which predicts exclusive kernels pixel by pixel and channel by channel dynamically conditioned on an input, boosting the model performance while decreasing the number of parameters. Experiments are conducted on five datasets: ShanghaiTech dataset, UCF_CC_50 dataset, UCF_QRNF dataset, GCC dataset and NWPU dataset and the ablation studies are performed on ShanghaiTech dataset. The final results show that the proposed DLMP-Net can effectively overcome the problems mentioned above and provides high crowd counting accuracy with smaller model size than state-of-the-art networks.

Keywords: Crowd density estimation · Feature fusion · Dynamic bottleneck block

S. Yu et al. (Eds.): PRCV 2022, LNCS 13537, pp. 27–39, 2022.
https://doi.org/10.1007/978-3-031-18916-6_3

1 Introduction

In crowd analysis, crowd density estimation is an important branch which can predict the density maps of congested scenes. This development follows the demand of real-life applications since the same number of people could have completely different crowd distributions, thus just counting the number in a crowd is not enough. The density map can help us obtain more accurate and comprehensive information and is essential for making correct decisions in high-risk environments such as violence and stampede. The recent development of crowd density estimation relies on CNN-based methods because of the high accuracy they have achieved in image classification, semantic segmentation and object detection. Though many compelling CNN-based models [1,2,4,6–8] have been proposed, it is still challenging to achieve high-precision crowd counting results in complex crowd scenarios in [2–5,10].

Due to different surveillance distances from the camera, people in images or across scenes usually exhibit various scales and distributions. To tackle these problems, previous methods mainly use multi-column network such as [1,2,4,7] or multi-dilated decoders such as [6,8] to extract image multi-scale features. Though these methods have achieved success, they still have limitations. On the one hand, multi-column (usually three columns) network usually introduces more redundant information leading to low efficiency. Especially for the scene that is over-crowded or over-sparse, roughly dividing head size into three levels is equivocal for the targets at the edge of the boundary. On the other hand, prevailing density estimation models based on multi-dilated convolution just simply uses one or two different dilation rate convolutions, which will not fully meet the requirement of scale variation.

Typical backbones for crowd density estimation are VGG or ResNet. Based on these backbones, dozens of crowd density estimation methods have been proposed and they have achieved some success, but there is still much big promotion space. The fundamental assumption in the design of these classic network layers is the same and static convolution kernels shared by all images in a dataset, which ignores the content diversity and introduces high memory consumption. To increase the content-adaptive capacity of a model, many groundbreaking dynamic filters [18,19,21,22] have been proposed. Even though they attain accuracy gains, they are either compute-intensive or memory-intensive, leading to a difficulty of model deployment on low-resource devices. To facilitate model deployment, brilliant model compression approach channel pruning [17] is reported to reduce the computation and storage cost, but improper application of this method may lead to performance drops. Thus, how to design a crowd density estimation network that can balance model accuracy and model size is a challenge. In this paper, we propose a dynamic yet lightweight multi-pyramid network (DLMP-Net) for crowd density estimation. It mainly consists of two parts: a ResNet backbone based on dynamic bottleneck block and a shuffle-pyramid feature fusion module. The architecture of the proposed network can be seen in Fig. 1. Unlike the most previous networks that use VGG16 as the backbone, we employ a ResNet101 backbone as the encoder for its stronger representation power. We substitute all original bottleneck blocks with

the proposed dynamic bottleneck block (DBB) in ResNwt101 to obtain content-diversity information. Then, a shuffle-pyramid feature fusion module (SPFFM) is designed for feature fusion to solve effectively the scale variation problem.

In general, the contributions of our work are twofold:

1) We present SPFFM to improve the feature fusion effectiveness and efficiency for crowd density estimation under complex scenes. The proposed SPFFM uses multi-dilated kernels and shuffled parallel group convolution to enlarge the receptive field and simultaneously improve the model inference speed.
2) We present DBB to achieve dynamic feature extraction depending on the inputs for crowd density estimation. The proposed DDB adjusts the convolution kernels dynamically conditioned on an input, which can attain richer semantic information requiring fewer parameters.

2 Related Work

For crowd density estimation, the challenge of scale change is one of the most important factors that affects model accuracy. To remedy this issue, MCNN [1] employs a multi-column convolutional neural network that utilizes multi-size filters to extract features for different receptive fields. Switch-CNN [2] further adds a density level classifier in front of the MCNN columns and allocates image patches with different density levels to corresponding branches to generate density maps. Analogously, CP-CNN [11] captures multi-scale information by using different CNN networks instead of multi-column structure to combine global and local context priors. To achieve better feature fusion, MBTTBF [20] presents a multi-level bottom-top and top-bottom fusion network to combine multiple shallow and deep features. Contrary to above multi-columns network structures, CSRNet [6] is a representative single-column density estimation network which utilizes six dilated convolution layers to expand the field of view. To go a step further, SFCN [7] adds a spatial encoder to the top of the FCN backbone, so as to improve the accuracy of population density map estimation. Though these works have achieved accuracy improvement by using different strategies of multi-scale feature fusion, they suffer from the increase of model complexity and parameters to some extent.

Therefore, lightweight networks become very popular in practical applications and model compression offers reduced computation cost. Channel pruning [17] as a typical model compression approach aims to remove the useless channels for easier acceleration in practice. Furthermore, other compact models such as Xception [12] and MobileNets [14] utilize lightweight convolutions to reduce the number of parameters. ShuffleNets [16] introduce channel shuffle operation to improve the information flow exchange between groups. Although these model compression approaches can effectively reduce the number of parameters, they usually sacrifice the model accuracy. In contrast to model compression approaches, dynamic convolution can effectively improve model accuracy. One of the early dynamic filters CondConv [18] predicts coefficients to combine

several expert filters and learns specialized convolutional kernels for different inputs. DynamicConv [19] follows the idea of CondConv [18] but involves sum-to-one constraint and temperature annealing for efficient joint optimization. Albeit these approaches have improved model accuracy, the combined filters require a large number of parameters. For better parameter-accuracy tradeoff, the latest dynamic filter DDF [21] decouples dynamic filters into spatial and channel ones, which attains content-adaption but is lightweight even compared with the standard convolutions. Involution [22] breaks through existing inductive biases of convolution and also achieves superior performance while reducing the model size and computation load. These two latest dynamic filters raise the curtain of dynamic yet lightweight network design.

3 Proposed Solution

The architecture of the proposed DLMP-Net is illustrated in Fig. 1. Inspired by the idea in [4], we choose ResNet101 as the backbone for its flexible architecture and stronger feature extraction ability. In order to ensure the consistency and fairness with other crowd density estimation networks, we only use the first three layers of the ResNet101 and change the stride of the third layer from 2 to 1 to preserve the scale of the final density maps. This is because if we continue to stack more convolutional layers and pooling layers, the output size would be further shrunken and it is hard to generate high-quality density maps [6]. In particular, we use the proposed DBB to replace the traditional bottleneck blocks in the ResNet101 to achieve dynamic feature extraction depending on the inputs. For the encoding stage, we propose a shuffle-pyramid feature fusion module (SPFFM) to enlarge the receptive field of the extracted feature maps, which outputs high-quality density maps and simultaneously improves the inference speed of our model.

3.1 Shuffle-Pyramid Feature Fusion Module

In crowd density estimation tasks, standard convolution has two main defects. One is that it usually has a fixed receptive filed thus cannot efficiently extract multi-scale features in crowd images. The other is that stacking different scales of standard convolution kernels increases of the number of parameters. To solve the above problems, we design SPFFM to capture multi-scale features in crowd scenes. Specifically, we first divide input feature maps into multiple blocks, and then in each divided block, we perform group convolution and different rates of dilated convolutions. The structure of SPFFM is shown in Fig. 1.

In each layer of SPFFM, the number of groups G follows a sequential increment by 2^n ($G = 2^0, 2^1, 2^2, 2^3$) in a pyramid-shape and the dilation rate r increases by 1 (r = 1, 2, 3 and 4) also following the sequential increment of groups in a pyramid-shape. For example, in the first layer, we assume that the channels of the input feature maps are K; We divide the input feature maps into four blocks, and each block contains C1, C2, C3, C4 channels respectively, thus

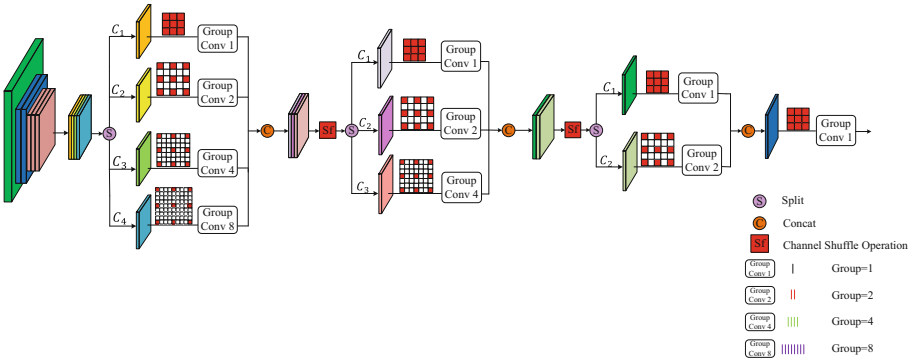

Fig. 1. The pipeline of the proposed DLMP-Net. We use ResNet101-DBB as the backbone. The shuffle-pyramid feature fusion module is used to extract multi-scale features. The detailed SPFFM is illustrated above.

apparently C1 + C2 + C3 + C4 = K. The first block contains C1 channels and the dilation rate and group number are 1 and 2^0, respectively. The second block contains C2 channels and the dilation rate and group number are 2 and 2^1, respectively. The third block and the last block continually follow the pyramid-shape increment as defined above. The second layer is divided into three blocks, the third layer is divided into two blocks and for the last layer, there is only one block in which we use standard convolution with G=1. Moreover, we introduce a channel shuffle operation before each division to improve the feature extraction efficiency and accuracy, since group convolution can reduce the amount of calculation. But between different groups, there is no connection and information exchanges. Channel shuffle operation can realize direct communication between different groups and prevents same feature maps between different layers from being divided into the same block.

According to the principle of SPFFM, the input feature maps are defined as I and the output feature maps are defined as O, then

$$O_i(I) = \begin{cases} Gconv(I, B_i, G_i, d_i), & i = 1 \\ Sf(Gconv(O_1(I), B_i, G_i, d_i)), & i = 2 \\ \quad\vdots \\ Sf(Gconv(O_{N-1}(I), B_i, G_i, d_i)), & i = N \end{cases} \tag{1}$$

where $Gconv(I, B_i, G_i, d_i)$ represents the group dilated convolution, B_i is the number of blocks, G_i is the number of groups, d_i is the dilation rate, and N is the number of layers. It is worth noting that B_i, G_i, d_i and N are hyperparameters. The Sf denotes the channel shuffle operation. The overall computational cost of SPFFM is

$$F(B, G, K, C_{in}, C_{out}) = \sum_{i=1}^{B} \left(\frac{K_i \times C_{in}^i \times C_{out}^i \times H \times W}{G_i}\right), \tag{2}$$

where F is the total computation cost, B is the divided blocks number and K is the size of the convolution kernel. C_{in}^{i} is the number of input features in the i_{th} group and C_{out}^{i} is the output feature maps number. G_i is the number of groups in the i_{th} block. It's obvious that dilated convolution can enlarge the receptive field while containing the resolution of the feature maps. Moreover, the group convolution can reduce overall computational load.

According to the aforementioned analysis, we can see that the proposed SPFFM not only provides better multi-scale feature representation, but also achieves faster inference speed and decreases the computational cost.

3.2 Dynamic Bottleneck Block

The deep CNNs based on vanilla convolution usually apply the same convolution kernels to all pixels in an crowd image, inevitably leading to sub-optimal feature learning. To address the above problem, we propose a dynamic bottleneck block, namely DBB, which adjusts the convolution kernels dynamically conditioned on an input.

In DBB, a spatial filter and a channel filter are respectively predicted depending on the input features through their corresponding filter prediction branches, then the two filters are pixel-wise multiplied to obtain a new filter as shown in Fig. 2. The new filter is finally used in image feature extraction. This operation can be written as:

$$V'_{(r,i)} = \sum_{p_n \in R} \left\{ D_i^{spatial} \cdot (p_i - p_j) \cdot D_r^{channel} \cdot (p_i - p_j) \cdot V_{(r,j)} \right\} \tag{3}$$

where $V'_{(r,i)}$ is the value at i_{th} pixel and r_{th} channel of output feature. $V_{(r,j)}$ is the value at j_{th} pixel and r_{th} channel the input feature. $D_i^{spatial}$ is the spatial dynamic filter at i_{th} pixel; $D_r^{channel}$ is the channel dynamic filter at r_{th} channel.

In practical applications, we usually consider DBB as a block and apply it to a backbone. For the prediction branch of spatial filter, the channel number is changed into K^2 only by 1×1 convolution, and the K^2 (which is reshaped as $K \times K$ later) corresponding to each pixel is what we want as a spatial filter. For the prediction branch of channel filter, a structure similar to SE attention operation [15] is adopted. Firstly, input feature maps are directly squeezed into a $C \times 1 \times 1$ tensor through global average pooling, and this tensor is considered to have global information. Then through a gate mechanism composed of two layers of fully connection (excitation layers), $C \times 1 \times 1$ is reshaped into $C \times K^2$, that is, each channel C_i corresponds to an attention value $(K^2)_i$. Then we combine the predicted spatial and channel filters by pixel-wise multiplication to obtain a new filter. Thus, the final filter is exclusive pixel by pixel and channel by channel. As the generated filter values might be extremely large or small for some input samples, the Filter Normalization is introduced to maintain the stability of the training process. Due to the limited space, we do not make the detailed explanation. We compared the performance of traditional ResNet101 and ResNet101-DBB in crowd density estimation, as shown in Fig. 3, our ResNet101-DDB is more sensitive on both sparse and crowded regions.

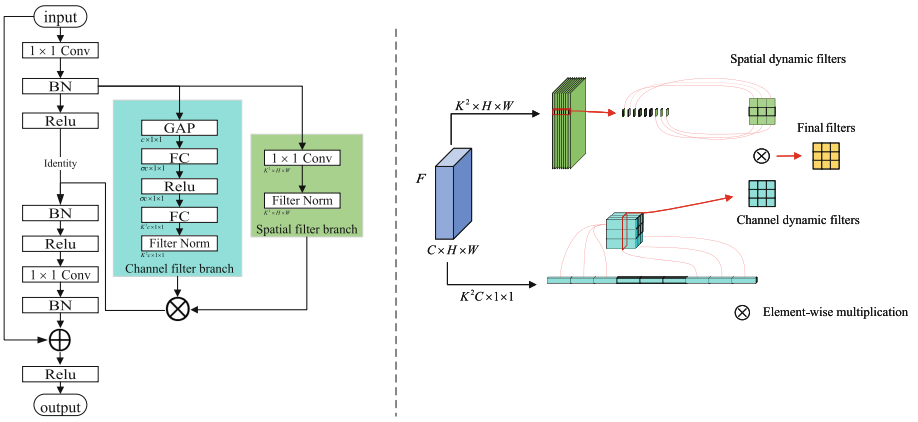

Fig. 2. The proposed Dynamic Bottleneck Block (DBB) and the detailed spatial/channel filter prediction branches.

Here we have a brief discussion about the parameter comparison between spatial/channel filters prediction branches in DDB and vanilla convolution. We assume n as the number of pixels, c as the channel numbers (for simplicity, both input channel and output channel numbers are c), k as the filter size and σ as the squeeze ratio in excitation layer. For prediction branch of spatial filter, the channel number is changed from c into k^2 only by 1×1 convolution and thus it contains ck^2 parameters; For prediction branch of channel filter, in the squeeze layer, it contains σc^2 parameters. In the excitation layer, the channel number is changed from σc into $k^2 c$, so it contains $\sigma c^2 k^2$ parameters. In total, the prediction branches contain $(ck^2 + \sigma c^2 + \sigma c^2 k^2)$ parameters. Considering that the values of k, c and σ, are usually set to 3, 256 and 0.2, the number of parameters for the prediction branches in DBB can be much lower even than a vanilla convolution layer.

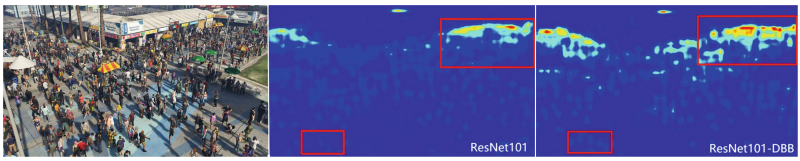

Fig. 3. The heat map comparison between ResNet101 and proposed ResNet101-DBB in the same crowd scene from GCC dataset.

4 Experiments

To investigate the effectiveness of the proposed DLMP-Net, we performed it on five popular crowd datasets: ShanghaiTech, UCF_CC_50, UCF-QRNF, GCC and NWPU-Crowd and compared with recent seven state-of-the-art crowd estimation models. We also illustrate the implementation details and evaluation metrics. In the last part of this section, we perform the ablation studies on the Shanghai Tech dataset.

4.1 Implementation Details

For data augmentation of training data, we use random cropping and horizontal flipping. For ShanghaiTech Part A and Part B, the crop size is 256×256 and 512×512 for other datasets. As for the groundtruth generation, we adopt the same strategy in [1], which utilizes Gaussian kernels to blur and smooth the head annotations. It is defined as:

$$F_i^{Groundtruth} = \sum_{x_i \in p} G_{\sigma^2}(x) \times \theta(x - x_i) \tag{4}$$

where the pixel position in the image is defined as x and the position of the i_{th} head on the annotation map θ is defined as x_i. $G_{\sigma^2}(x)$ is the Gaussian kernel and σ is the deviation of Gaussian distribution. We set the Gaussian kernel size to 15 and σ to 4 for all datasets for fair comparison. The backbone is pretrained on the ImageNet dataset [9] and we use the Adam algorithm with a learning rate 1×10^{-5} for optimization. Our DLMP-Net is implemented on an eight NVIDIA RTX 3090 GPU (24 GB).

4.2 Evaluation Metrics

For crowd density estimation, we use two evaluation metrics namely mean absolute error (MAE) and mean squared error (MSE).

$$MAE = \frac{1}{N} \sum_{i=1}^{N} \left| Y_i - \widehat{Y_i} \right| \tag{5}$$

$$MSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left| Y_i - \widehat{Y_i} \right|^2} \tag{6}$$

where N is the samples number and Y_i is the groundtruth crowd counting result of the i^{th} image. $\widehat{Y_i}$ is the predicted count of the i^{th} image. We sum up the predicted density map to obtain the counting result. To measure the quality of the predicted density map, we choose the pixel-wise mean square error (MSE) loss

as the objective function. During the optimization process, the model parameter β is defined as follows:

$$Loss\,(\beta) = \frac{1}{2Z}\sum_{i=1}^{Z} ||S_i^{GT} - \hat{S}_i^{PRE}||_2^2 \tag{7}$$

where batch size is Z and the groundtruth density map of the input image is S_i^{GT}. The estimated density map is $\hat{S}_i^{PRE}$.

4.3 Main Comparison

We compared our method with seven state-of-the-art methods on five popular datasets comprehensively. The overall comparison results demonstrate that our DLMP-Net can provide good prediction results for various complex scenarios.

ShanghaiTech Dataset. In ShanghaiTech Part A, 300 and 183 images are used for training and testing, respectively. The ShanghaiTech Part B includes 716 images, in which 400 images are used for training and 316 images are used for testing. It can be seen in Table 1 that our DLMP-Net obtains MAE/MSE of 59.2/90.7 as the best performance.

UCF_CC_50 Dataset. UCF_CC_50 dataset includes 50 images with different perspectives and resolutions, and we follow the 5-fold cross-validation method in [13]. In the experiments, four groups are used for training and the last one group is used for testing. As shown in Table 1, Our DLMP-Net still reached the smallest values of MAE/MSE of 183.7/268.5. Notably, the overall MSE/MAE values are much higher in all approaches compared with others datasets because UCF_CC_50 dataset contains imbalanced samples and small number of images.

UCF_QRNF Dataset. The images are divided into the training set with 1,201 images and the test set with 334 images, respectively. As shown in Table 1, our DLMP-Net attains optimal MAE/MSE values of 99.1/169.7, especially the MAE value is less than 100, which shows our stronger feature extraction ability.

GCC Dataset. GCC consists of 15,212 images and we randomly divided 75% images for training and rest 25% for testing. Table 1 shows the comparison between our DLMP-Net and seven other state-of-the-art approaches. Our DLMP-Net attains the smallest values of MAE/MAE, namely 25.3/73.2, especially the MSE value achieves a tremendous drop from nearly 80 to 66.2.

NWPU-Crowd Dataset. The NWPU-Crowd dataset is a large-scale and challenging dataset that contains 5,109 images and 2,133,238 labeled instances. Table 1 shows that the value of MAE for DLMP-Net is 87.7, with an improvement of 7.3 over SFCN† [7]. The value of MSE provided a slight reduction of

Table 1. Experimental results of DLMP-Net with other seven state-of-the-art methods on five main stream datasets. The best values are in bold.

Methods	ShanghaiTechA		ShanghaiTechB		UCF_CC_50		UCF_QRNF		GCC(RS)		NWPU-crowd	
	MAE(↓)	MSE(↓)	MAE(↓)	MSE(↓)	MAE(↓)	MSE(↓)	MAE(↓)	MSE(↓)	MAE(↓)	MSE(↓)	MAE(↓)	MSE(↓)
CSRNet (2018)	68.2	115.0	10.6	16.0	266.1	397.5	121.3	208.0	38.5	86.6	103.0	433.8
SCAR (2018)	66.3	114.1	9.5	15.2	259.0	374.0	-	-	31.7	76.8	-	-
SFCN (2019)	67.0	104.5	8.4	13.6	266.4	397.6	135.1	239.8	36.1	81.0	106.2	615.8
SFCN† (2019)	64.8	107.5	7.6	13.0	245.3	375.8	114.5	193.6	28.8	71.2	95.0	597.4
CAN (2019)	62.3	100.0	7.8	12.2	212.2	243.7	107.0	183.0	-	-	-	-
PSCC+DCL (2020)	65.0	108.0	8.1	13.3	-	-	108.0	182.0	31.3	83.3	-	-
HYCNN (2020)	60.2	94.5	7.5	12.7	184.4	270.1	100.8	185.3	-	-	-	-
Our DLMP-Net	**59.2**	**90.7**	**7.1**	**11.3**	**183.7**	**268.5**	**99.1**	**169.7**	**25.3**	**73.2**	**87.7**	**431.6**

2.2 compared with CSRNet [6], this is because crowd scenes in NWPU-Crowd have a various distribution. Figure 4 further shows the quantitative counting results and comparative estimated density maps on the NWPU-Crowd dataset using SCAR [8], SFCN† [7], and our DLMP-Net. It is clear that the proposed DLMP-Net can provide better counting results and high quality density maps for various complex scenarios.

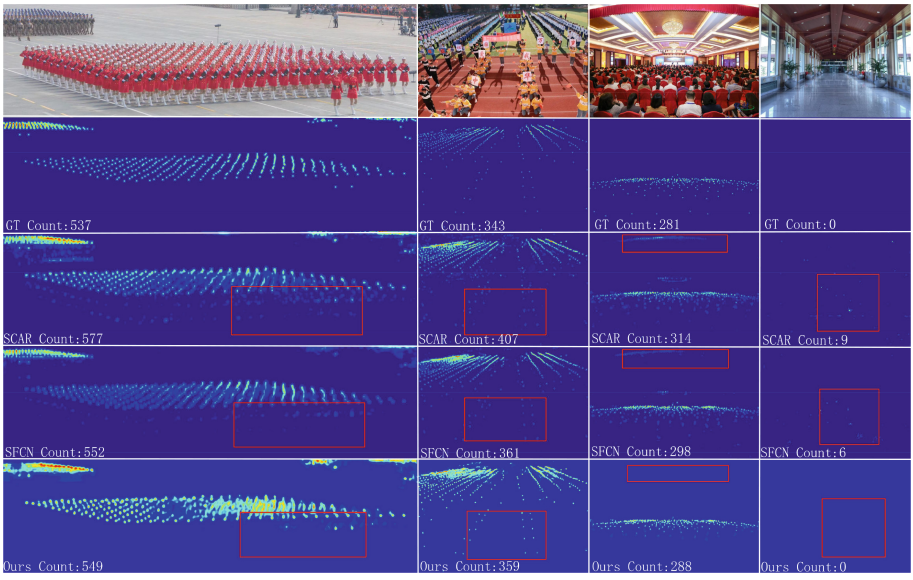

Fig. 4. The density estimation results comparison between SCAR, SFCN and DLMP-Net on NWPU dataset

4.4 Ablation Studies

To demonstrate the effectiveness of our DLMP-Net, we perform ablation experiments on ShanghaiTech dataset and the results are presented in Table 2. For all ablation studies, we adopt ResNet101 as the basic component and combine three different types of SPFFM with or without DDB to further investigate the effectiveness. Specifically, SPFFM (A) utilizes vanilla convolution with the kernel size 3×3, SPFFM(B) utilizes dilated convolution with fixed dilation rate r = 2 and SPFFM (Ours) is the proposed method with the dilation rate r =(1, 2, 3 and 4). It can be seen that without DBB, all model sizes increase 8.5M and using vanilla convolution for feature fusion has the worst performance. Compared with vanilla convolution and fixed-rate dilated convolution, SPFFM (ours) has tremendously improvements of 24.6/55.2 (MAE/MSE) over SPFFM(A) and 10 /20.6 (MAE/MSE) over SPFFM(B) on ShanghaiTechA even without adding DBB. On ShanghaiTechB, SPFFM (ours) also achieves the best results. In the last three ablation experiments, we combined DBB with three different types of SPFFM. The experimental results show that we can obtain more accurate and efficient models by adding DBB. SPFFM (ours) + DBB greatly improves 20.2/45.3 (MAE/MSE) over SPFFM(A)+DBB and 8.6/24.2 (MAE/MSE) over SPFFM(B)+DBB on ShanghaiTechA. Significant improvements can also be seen on ShanghaiTechB.

Table 2. Comparison of the model size and MAE/MSE on ShanghaiTech dataset for different SPFFM types and DBB. The best values are in bold.

backbone	SPFFM structure			DBB	Parameters (M)	ShanghaiTechA		ShanghaiTechB	
	SPFFM (A)	SPFFM (B)	SPFFM (Ours)			MAE(↓)	MSE (↓)	MAE (↓)	MSE (↓)
ResNet101	✓				33.27	87.2	157.0	28.4	48.4
		✓			33.27	72.6	122.4	12.2	17.6
			✓		33.27	62.6	101.8	7.7	12.9
	✓			✓	24.77	79.7	136.0	15.1	21.8
		✓		✓	24.77	67.8	104.9	10.2	16.5
Our DLMP-Net			✓	✓	**24.77**	**59.2**	**90.7**	**7.1**	**11.3**

5 Conclusion

In this paper, we present a dynamic yet lightweight multi-pyramid network (DLMP-Net) for crowd density estimation. DLMP-Net is superior to the current models due to the design of a shuffle-pyramid feature extraction and fusion module (SPFFM) and a dynamic bottleneck block (DBB) which can replace the standard bottleneck blocks in ResNet backbone. SPFFM employs multi-dilated convolution to extract and fuse various scale features. In addition, we add group and channel shuffle operation to reduce the model complexity and improve the efficiency of feature fusion. DBB predicts exclusive kernels pixel by pixel and channel by channel dynamically conditioned on the input, boosting the model performance while decreasing the number of parameters. Extensive experiments

demonstrate that the proposed DLMP-Net has better performance for crowd counting and density estimation in different congested images. In future work, we will further investigate the improved dynamic filters and explore how to form a fully dynamic crowd density estimation network.

Acknowledgements. This work was supported in part by Natural Science Basic Research Program of Shaanxi (Program No. 2022JQ-634).

References

1. Zhang, Y., Zhou, D., Chen S., Gao, S., Ma, Y.: Single-image crowd counting via multi-column convolutional neural network. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 589–597. IEEE, Las Vegas (2016)
2. Sam, D.B., Surya, S., Babu R.V.: Switching convolutional neural network for crowd counting. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5744–5752. IEEE, Hawaii (2017)
3. Wang, Q., Gao, J., Lin, W., Li, X.: A large-scale benchmark for crowd counting and localization. IEEE Trans. Pattern Anal. Mach. Intell. **1**(3), 7 (2020)
4. Wang, Q., Gao, J., Lin, W., Yuan, Y.: Learning from synthetic data for crowd counting in the wild. In: 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8198–8207. IEEE, Los Angeles (2019)
5. Idrees, H., et al.: Composition loss for counting, density map estimation and localization in dense crowds. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11206, pp. 544–559. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01216-8_33
6. Li, Y., Zhang, X., Chen, D.: CSRNet: dilated convolutional neural networks for understanding the highly congested scenes. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1091–1100. IEEE, Salt Lack City (2018)
7. Liu, W., Salzmann, M., Fua, P.: Context-aware crowd counting. In: 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5099–5108. IEEE, Los Angeles (2019)
8. Gao, J., Wang, Q., Yuan, Y.: SCAR: spatial-channel-wise attention regression networks for crowd counting. Neurocomputing **363**, 1–8 (2019)
9. Deng, J., Dong, W., Socher, R.: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 248–255. IEEE, Miami (2009)
10. Idrees, H., Saleemi, I., Seibert, C., Shah, M.: Multi-source multi-scale counting in extremely dense crowd images. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2547–2554. IEEE, Portland (2013)
11. Sindagi, V.A., Patel, V.M.: Generating high quality crowd density maps using contextual pyramid CNNs. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 1861–1870, IEEE, Venice (2017)
12. Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1251–1258. IEEE, Hawaii (2017)
13. Lei, T., Zhang, D., Wang, R., Li, S., Zhang, Z., Nandi, A.K.: MFP-Net: multi-scale feature pyramid network for crowd counting. IET. Image Process. **15**(14), 3522–3533 (2021)

14. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T.: MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861. (2017)
15. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7132–7141. IEEE, Salt Lack City (2018)
16. Ma, N., Zhang, X., Zheng, H.-T., Sun, J.: ShuffleNet V2: practical guidelines for efficient CNN architecture design. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018. LNCS, vol. 11218, pp. 122–138. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01264-9_8
17. Wen, W., Wu, C., Wang, Y., Chen, Y., Li, H.: Learning structured sparsity in deep neural networks. In: Advances in Neural Information Processing Systems (NIPS), vol. 29 (2016)
18. Yang, B., Bender, G., Le, Q.V., Ngiam, J.: Condconv: conditionally parameterized convolutions for efficient inference. Adv. Neural Inf. Process. Syst. (NeurIPS) **32**, 1–12 (2019)
19. Chen, Y., Dai, X., Liu, M., Chen, D., Yuan, L., Liu, Z.: Dynamic convolution: attention over convolution kernels. In: 2020 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11030–11039. IEEE, Seattle (2020)
20. Sindagi, V.A., Patel, V.M.: Multi-level bottom-top and top-bottom feature fusion for crowd counting. In: 2019 International Conference on Computer Vision, pp. 1002–1012, IEEE, Los Angeles (2019)
21. Zhou, J., Jampani, V., Pi, Z., Liu, Q., Yang, M.H.: Decoupled dynamic filter networks. In:2021 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6647–6656. IEEE, Online (2021)
22. Li, D., Hu, J., Wang, C., Li, X., She, Q., Zhu, L.: Involution: inverting the inherence of convolution for visual recognition. In:2021 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12321–12330. IEEE, Online (2021)