Future Generation Computer Systems 147 (2023) 328–338

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Crowd counting in smart city via lightweight Ghost Attention Pyramid Network

Xiangyu Guo [a], Kai Song [b], Mingliang Gao [a,*], Wenzhe Zhai [a], Qilei Li [c], Gwanggil Jeon [a,d,**]

[a] School of Electrical and Electronic Engineering, Shandong University of Technology, Zibo, 255000, China
[b] Network Management and Security Protection Room, Information Service Center, Zhengzhou, Henan, 450053, China
[c] School of Electronic Engineering and Computer Science, Queen Mary University of London, London, E1 4NS, United Kingdom
[d] Department of Embedded Systems Engineering, Incheon National University, Incheon 22012, Republic of Korea

ARTICLE INFO

Article history:
Received 21 November 2022
Received in revised form 19 April 2023
Accepted 13 May 2023
Available online 23 May 2023

Keywords:
Smart city
Crowd counting
Lightweight network
Scale variation
Attention mechanism

ABSTRACT

Crowd counting targets for determining the number of pedestrians in an image, which is of crucial importance for smart city construction. The problem of scale variation is an ingrained and drastic challenge in crowd counting, and it severely degrades the performance of counting. To address this problem, many powerful models with complex network structures and tricks are built, but the constrained resources of embedded systems prevent the direct deployment of these models into an edge device. Thus, it is on high demand to design favourable lightweight models that require fewer parameters and a fast inference speed, while maintaining competitive counting performance. To this aim, we devise a lightweight network, termed as Ghost Attention Pyramid Network (GAPNet). Specifically, a lightweight GhostNet is adopted as the backbone to encode low-level features. Subsequently, a zero-parameter channel attention module is designed to select the discriminative crowd region efficiently. In addition, an efficient pyramid fusion module is built with a four-branch architecture to obtain multiscale hierarchy representations while reducing the parameters. Finally, a decoder generates the prediction by exploiting a series of transposed convolution blocks. Extensive experiments on crowd counting benchmarks have proved the superiority of the GAPNet in both accuracy and efficiency.

1. Introduction

In recent years, the impact of urbanization has put huge pressure on city management and planning. The emergence of Internet of Things devices and miniaturized sensing technologies have promoted the progress of smart cities. Specifically, crowd analysis plays a crucial role in many smart city applications, *e.g.,* crowd tracking [1], crowd counting [2], and drone-based crowd analysis [3]. Among the applications in crowd analysis, crowd counting is a fundamental and practical task that targets for inferring the number of pedestrians in a still image or video sequence. The crowd counting task can be applied to a lot of real-world scenarios, such as large gatherings, urban planning and pedestrian monitoring [2,4].

Crowd counting algorithms can be divided into three categories: detection-based [5], regression-based [6] and density estimation-based methods [7–9]. Specifically, the density estimation-based method relies on the powerful feature extraction ability of convolutional neural network (CNN) to regress a density map, and then sum the pixels on the density map to get the counts. It is far more accurate and stable than the other two methods and has become the mainstream counting approach.

Nevertheless, most of current methods always adopt bulky networks as the backbone, *e.g.,* VGG and ResNet, to extract features or design complex modules to refine features. No gainsaying, the counting results with dense networks are excellent, but the results are achieved at the cost of large resource consumption, *e.g.,* computational time and GPU memory.

Fig. 1 shows the comparison of parameters and counting accuracy of some competitive models on Shanghai Part A dataset (MCNN [10], TDF-CNN [11], SANet [12], CAN [13], BL [14], SASNet [15], UEPNet [16] and ours). It proves that superior counting performance is often accompanied by an increase in parameters. For example, the SASNet outperforms the proposed method in MAE. Because the SASNet is a network with huge parameters. The parameter in SASNet is 38.9M, which is 13.75 times the parameters of GAPNet (2.85M). However, a large number of parameters will undoubtedly lead to a decrease in counting efficiency, which limits the deployment of the model in real scenarios. Therefore, it is a challenging and profound task to solve the problem of scale variation under the premise of the lightweight network scheme.

* Corresponding author at: School of Electrical and Electronic Engineering, Shandong University of Technology, Zibo, 255000, China.
** Corresponding author at: Department of Embedded Systems Engineering, Incheon National University, Incheon 22012, Republic of Korea.
E-mail addresses: mlgao@sdut.edu.cn (M. Gao), ggjeon@gmail.com (G. Jeon).

https://doi.org/10.1016/j.future.2023.05.013
0167-739X/

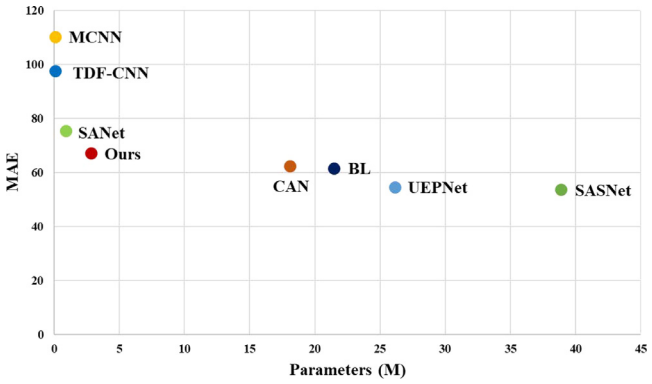

Fig. 1. Comparison of the parameters and accuracy of different models on Shanghai Part A dataset. Low MAE indicates high counting accuracy.

To this aim, several lightweight networks [17–19] have been introduced to boost the counting efficiency. Liu et al. [18] proposed a knowledge transfer mechanism to train an efficient "student" network through a well-trained "teacher" network. Gao et al. [19] adopted some compact convolution blocks to tackle the appearance similarity and attained competitive counting performance. Despite the efficiency gains, there are still some defects. The first and most important point is the unconvincing counting performance. The purpose of crowd counting is to count the number of people as precisely as possible, but poor counting results against this aim. Because the most direct way to reduce the number of parameters is to remove numerous convolution layers, the extraction of features with only a few convolutional filters is yet insufficient. Zhang et al. [10] adopted three branch convolution layers to predict the counts and there are few parameters, but it lacks accuracy. SANet [12] was built with a series of convolution filters for efficient crowd counting, but the counting results are also unsatisfactory as it fails to extract low-level features. Wang et al. [17] employed a lightweight network MobileNet as encoder, and designed simple fusion blocks to obtain multiscale information. Despite the support of the backbone network, the counting accuracy is not ideal.

Another point is that extracting multiple hierarchy information has always been a challenge in the field of crowd counting, especially in lightweight networks. Restricted convolution layers cannot meet the demand of encoding low-level features and rich spatial semantic information. To cope with the problem, multiscale feature fusion is an effective way, which mainly built a pyramid structure to extract features with different scales. SASNet [15] designed a density head and confidence head branch to adaptively select the crowd region with different density levels. STNet [20] proposed a scale tree network to extract hierarchical features, which was capable of boosting the scale diversity of density map. Furthermore, some works [21,22] introduced attention modules to select the crowd region with different density levels. Guo et al. [22] proposed a dilated attention module to generate different interest maps, which consist of rich scale information.

Despite the protracted efforts, the problem of scale variation is still the most difficult problem to solve in a crowd scenario, especially in a dense crowd scenario. Compared with the sparse crowd scenarios, the scale variations in dense scenarios are more drastic and the CNN-based models proposed previously are difficult to capture more scale-level features due to the limited receptive field. Certainly, the scale variations can be addressed by enlarging the receptive field, but it will increase the complexity of the model. To this aim, the motivation of the paper is to solve the problem of scale variation under the premise of the lightweight network scheme.

In this work, we propose the Ghost Attention Pyramid Network (GAPNet) to address the problem of scale variations in crowd counting under the premise of the lightweight network scheme. Specifically, we deploy the GhostNet [23] as the encoder to extract low-level representations. Then, a zero-parameter channel attention (ZCA) module and an efficient pyramid fusion (EPF) module are proposed to handle the scale variation. Finally, several transposed convolution layers are stacked as the decoder to output the prediction. The ZCA module is effective and efficient because it needs no parameters to be learned, only through a series of simple operations, *i.e.,* average pooling, linear transforms and activation function, can adjust the weight of the channels. Subsequently, the EPF module is built with a four-branch pyramid structure to extract multiscale features. Each branch is composed of group convolutions and dilated convolutions, which is helpful to reduce the number of parameters. In short, this paper makes the following contributions:

1. A lightweight network termed GAPNet is proposed for crowd counting. It can be adapted to different scales and accomplish counting tasks efficiently and accurately.
2. A ZCA module is built to select the crowd region. Meanwhile, an EPF module is proposed to capture multi-hierarchy information efficiently.
3. Extensive experiments are carried out to verify the performance of crowd counting in accuracy and efficiency. Meanwhile, detailed ablation studies are conducted to prove the effectiveness of the individual module in the proposed model.

The following sections are structured as follows. Section 2 reviews the works related to the proposed method. Section 3 introduces the proposed method in detail, and Section 4 makes experimental comparison and analysis. Finally, the paper is summarized in Section 5.

2. Related work

In this section, we revisit three types of network related to the proposed method, *i.e.,* multiscale fusion-based network, attention-based network and lightweight-based network.

2.1. Multiscale fusion-based network

The scale variation is an inherent and chronic challenge in the field of crowd counting. It has been hindering the enhancement of the counting accuracy. An effective way to solve the problem is multiscale information fusion [10,24].

Zhang et al. [10] constructed a multi-column network, using three different branches to obtain features with different scales. The output of each branch is concatenated to generate a high-quality density map. In order to fit the crowd density in crowded scenes, Liu et al. [25] utilized different dilated convolutional layers in the back-end, to deal with the scale variation. Similarly, Liu et al. [13] proposed a context-aware network (CAN) to encode the scale representations adaptively. Particularly, it adopts scale pyramid pooling to increase the scale diversity, then a geometry-guided context learning mechanism is introduced to adapt to the foreground context. Jiang et al. [24] built an attention scaling network to produce the scaling factors, which are employed to multiply with the density map to adjust to the diverse crowd region densities. The final crowd density map can be generated by summing each fine-grained feature maps. Song et al. [15] adopted the U-Net as backbone to obtain five representations with different sizes. Then, a density head branch and a confidence head branch are built to select the crowd region adaptively. Wang et al. [20] proposed a scale tree diversity enhancer

to amplify scale representations hierarchically. Then, a cross-scale communication is introduced to strengthen the correlation between adjacent scales.

However, multiscale information fusion requires constructing a multi-column architecture, which will result in structural bloat and numerous parameters consuming [2]. For this purpose, we design an EPF module to capture scale information, which is both effective and efficient.

2.2. Attention-based network

The attention mechanism can guide the model to focus on the foreground by adjusting the channel or spatial weights adaptively. Recently, a number of attention modules have been successfully applied to crowd counting with incredible results. Guo et al. [22] proposed a scale-aware attention fusion module to generate attention maps which concentrate on the discriminative crowd scenes. Sindagi et al. [26] designed a spatial attention module to recognize the foreground, which can be exploited to boost the feature response. Guo et al. [21] proposed an attention module in the frequency domain, which can get more frequency information to select the foreground. Furthermore, a spatial attention is introduced to stress the heads with different scales. Chen et al. [27] built a variational attention and an intrinsic variational attention to guide the network to handle the specific domains. Wang et al. [28] incorporated an attention layer in the back-end to produce a segmentation density map. The segmentation map can save useful context information and guarantee accurate crowd region.

Undoubtedly, the foregoing methods can achieve the excellent counting performance, but the attention modules always come with some parameters to be learned. To this end, we propose a zero-parameter attention module to adjust the weight of the channel without parameter growth.

2.3. Lightweight-based network

In order to meet the requirements of practical engineering, the lightweight counting model has been explored by many scholars [11,17,19]. These models are expected to apply fewer parameters to get a satisfactory counting result.

The MCNN [10] is not only the first proposed network to increase scale diversity, but also a lightweight model. It only adopts several simple convolution filters to extract the context information. Cao et al. [29] utilized four convolution blocks as encoders to extract features, in which each convolution block is composed of four parallel convolution layers with different kernel sizes. Wang et al. [17] built a lightweight network which adopts MobilenetV2 [30] as the backbone to encode the scale representations, then several chained residual blocks and fusion blocks are applied as decoder to predict the density map. The encoder–decoder MobileCount can attain a balance between counting precision and efficiency. Gao et al. [19] discarded the process of extracting basic network from backbone, and directly designed a lightweight perspective crowd counting network. It consists of a density map estimation for local feature recognition, a random high-level density classification for global feature extraction, and a fore/background segmentation for identifying the crowd region. Sam et al. [11] proposed a top-down feedback network to refine the initial prediction. It is composed of two efficient modules, *i.e.,* bottom-up and top-down modules. The bottom-up module built two columns of CNNs, each consisting of convolutional layers with different receptive fields (9×9, 7×7, 5×5 and 3×3). The top-down module leverages the similar structure to generate the final density map. Ma et al. [31] built a hierarchical network called lightweight count network to address the crowd distribution prediction in congested scenes. Liu et al. [18] designed a Structure Knowledge Transfer (SKT) network to replace the dense backbone. Shi et al. [32] built a compact CNN with three parallel convolution filters for real-time crowd counting.

In this paper, we propose a lightweight network based on GhostNet for crowd counting, which ensures the counting performance and a few parameters.

Table 1
Architecture of the decoder in GAPNet.

Layers	Kernel size	Output channels	Activation function
Layer1	4×4	80	ReLU
Layer2	4×4	40	ReLU
Layer3	4×4	24	ReLU
Layer4	4×4	16	ReLU
Layer5	4×4	1	ReLU

3. Methodology

The architecture of the proposed GAPNet is illustrated in Fig. 2. It consists of four parts, an encoder for basic feature extraction, a ZCA module for selecting the crowd region, an EPF module for multiscale feature fusion, and a decoder for the density map prediction. Specifically, the encoder adopts the GhostNet [23] as the backbone, in which discards the final pooling and linear layer. Moreover, the decoder is composed of several transposed convolution blocks to resize the density map to the same size as the input (see Table 1).

3.1. Ghostnet

Lightweight networks require few convolutional filters and small convolution kernel sizes. To this end, we adopt the GhostNet [23] as the backbone to encode basic representations, which is able to discard plenty of redundant feature maps. Specifically, it is a stack of several Ghost bottleneck blocks which termed as Ghost module, as shown in Fig. 3.

Given the input $I \in \mathbb{R}^{C\times H\times W}$, a set of 1×1 convolution filters are first utilized to reduce the input channels by a half and produce the squeezed features $X \in \mathbb{R}^{\frac{C}{2}\times H\times W}$. Next, a cheap linear (CL) operation [23] is adopted to each channel of X to obtain ghost features $Y \in \mathbb{R}^{\frac{C}{2}\times H\times W}$. Specifically, the CL operation can be viewed as a group convolution. The number of groups is equal to the number of input channels. Note that the parameters of CL operation is much less than that of convolution operation, and the ghost features generated by linear operations can well replace redundant features generated by convolution filters. Finally, the output $O \in \mathbb{R}^{C\times H\times W}$ can be generated by a concatenation with X and Y. In a nutshell, the ghost module can be defined as,

$$O = \mathrm{Cat}(\mathrm{Conv}_1(I), \mathrm{CL}(\mathrm{Conv}_1(I))), \tag{1}$$

where Cat and CL denote the concatenation and cheap linear operations, respectively. Afterwards, two ghost modules are adopted to form the ghost bottleneck block, shown in Fig. 4. It is divided into two types according to the different strides. The bottleneck block with the stride of 2 can reduce the feature spatial dimension by half to capture the semantic information of relative details. Eventually, the GhostNet can be constructed by stacking ghost bottleneck blocks according to the structure of MobileNet [30].

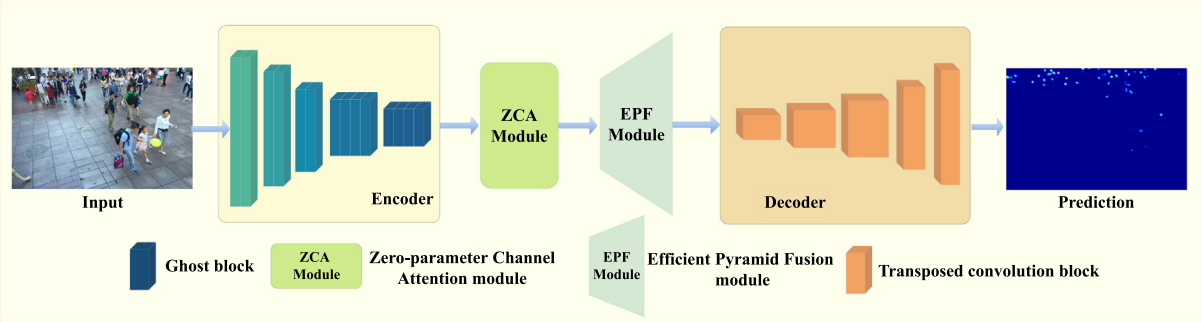

Fig. 2. Architecture of the proposed GAPNet. It consists of an encoder, a ZCA module, an EPF module and a decoder. The encoder is composed of stacked ghost blocks, and the decoder consists of a batch of transposed convolution blocks.

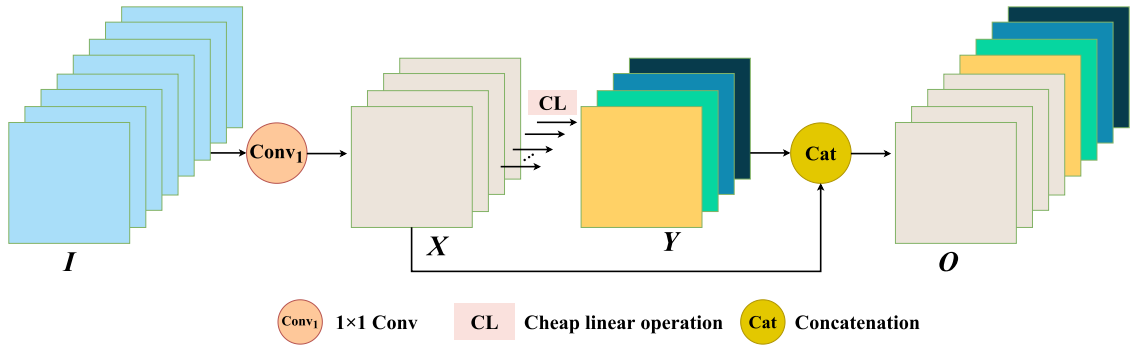

Fig. 3. Architecture of the Ghost module. First, convolution layers with a kernel size of 1 × 1 are employed for channel reduction. Then, a cheap linear operation is utilized to generate ghost feature maps. Last, a concatenation operation is executed to generate the output.

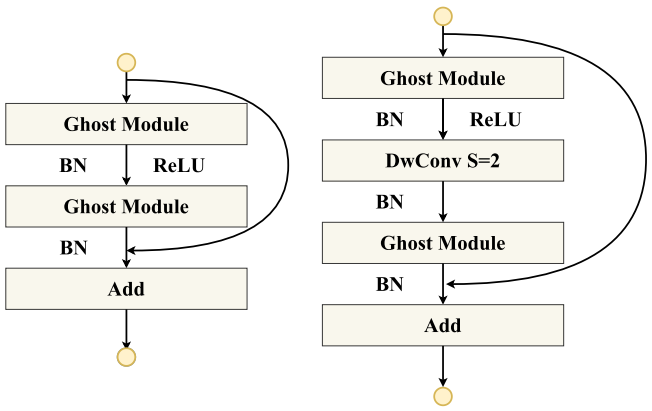

Fig. 4. Architecture of the ghost bottleneck block. 'BN' and 'ReLU' denote the batch normalization and ReLU activation function, respectively. 'DwConv' and 'Add' are the depth-wise convolution and addition operations.

3.2. Zero-parameter channel attention module

The objective of channel attention in crowd counting is to focus on the region where exists people from the feature map. Linear transform is able to increase the importance of target channels [33]. Specifically, it evaluates the linear separability between the target channels and other channels based on the energy function as,

$$E = \frac{4(\sigma^2 + \lambda)}{(f_c - \mu)^2 + 2\sigma^2 + 2\lambda}, \tag{2}$$

where μ and σ^2 denote the mean value and variance of the feature f_c, respectively. λ is a constant to prevent the σ from being 0, and it is set to 0.0001 in this paper. Eq. (2) states that the value of the energy function is determined by the linear transformation of the mean and variance of the spatial features of each channel. The lower the value of E, the more distinguishable the channel is from the surrounding channels, and the higher the weight should be given. In other words, the reciprocal of the Eq. (2) is proportional to the importance of the channel.

Based on the above discussion, we introduce the ZCA module to boost the counting performance without parameter growth. The architecture of the ZCA module is shown in Fig. 5.

Given an input $\mathbf{I}$, a global average pooling operation is first executed to reduce the size of $\mathbf{I}$ to 1×1, which is able to get a set of global attention maps X. Based on the energy function, the new channel attention maps Y are subsequently generated by performing a series of element-wise operations on X. It is computed as follows,

$$Y = \sum_{i=1}^{C} \frac{(X_i - \mu)^2 + 2(\sigma^2 + \lambda)}{4(\sigma^2 + \lambda)}, \tag{3}$$

where X_i represents the ith channel attention map. Following that, the re-adjust attention map Y is obtained by a Sigmoid function. Finally, $\mathbf{I}$ is multiplied by the Y to produce the output $\mathbf{O}$. In a nutshell, the function of the ZCA module can be formulated as,

$$\mathbf{O} = \mathbf{I} \times \{\mathrm{Sig}\langle f_e(\mathrm{GAP}(\mathbf{I}))\rangle\}, \tag{4}$$

where GAP and f_e are the global average pooling and linear transform operations, respectively. The ZCA module has no parameters to be trained, and the optimized channel feature map can be obtained only through some mathematical operations, which can effectively emphasize crowd areas.

3.3. Efficient pyramid fusion module

After the crowd area is selected by the ZCA module, the subsequent essential issue is to get rich spatial information to adapt

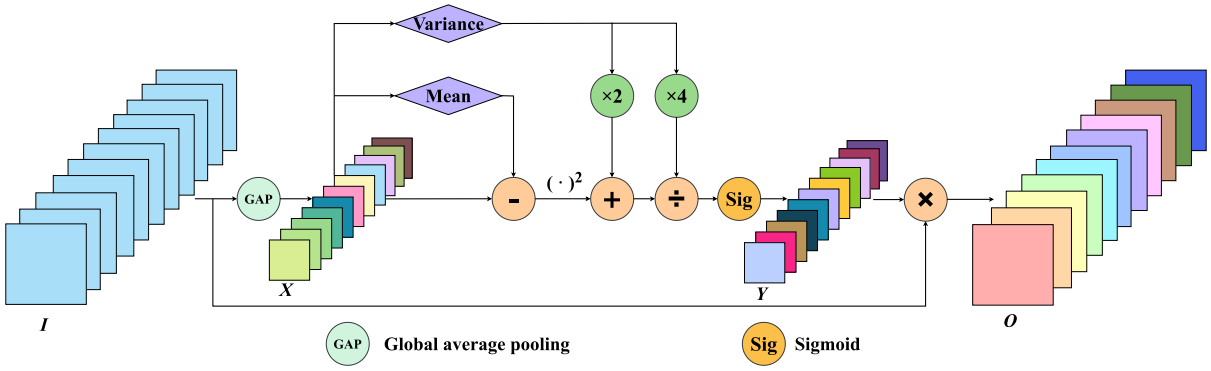

Fig. 5. Architecture of the proposed ZCA module. First, a global average pooling operation squeezes the spatial size of the input to 1 × 1. Then, a series of linear operations differentiate the importance of channels. Last, the input is multiplied by the weight to generate the optimized feature maps.

Table 2
Detailed information of the crowd counting datasets.

Dataset	Images	Avg.resolution	Train/Val/Test	Avg.count	Total.count	Annotation format
ShanghaiTech Part A [10]	482	589 × 868	300/ -/ 182	501	241,677	point-level
ShanghaiTech Part B [10]	716	768 × 1024	400/ -/ 316	123	88,488	point-level
UCF_CC_50 [34]	50	2101 × 2888	50/ 10/ 50	1,279	63,974	point-level
UCF-QNRF [35]	1,535	2013 × 2902	1,201/ -/ 334	815	1,251,642	point-level
WorldExpo'10 [36]	3,980	576 × 720	3,380/ -/ 600	50	199,923	point-level
NWPU-Crowd [37]	5,109	2191 × 3209	3,109/ 500/ 1,500	418	2,133,375	point&box-level
CARPK [38]	1,448	720 × 1280	989/ -/ 459	62	89,777	box-level

to the scale variation of the heads. To this end, an effective way is to construct a pyramid structure, and exploit receptive fields with different scales to extract spatial representations. However, most of the pyramid structures [13,20] generally adopt multi-branch structure, each branch adopting convolution filters of different sizes to learn spatial features. The structure is redundant and is not suitable for application in a lightweight network.

In order to solve the scale variation efficiently, we propose the EPF module, as shown in Fig. 6. It is a four-branch pyramid structure, and each branch extracts the features with different scales independently. At the back end of the module, a concatenation operation is executed to obtain rich spatial information. Compared with the aforementioned pyramid structures, the improvements of the EPF modules are two-folds. First, the traditional convolution is replaced by the group convolution. Group convolution divides the convolution filters into different groups, which are responsible for input at a particular depth, which ensures efficient training, simplification of the model and prevention of overfitting [23,30]. Second, dilated convolutions with fixed kernel size (3 × 3) instead of larger filters (*e.g.*,5 × 5, 7 × 7) are used to obtain rich spatial information. Dilated convolution has been widely applied to increase the scale diversity of the network without parameter increasing. The generated feature F_i of each branch is represented as,

$$F_i = \mathrm{Conv}_{3\times3}(d_i, G_i)(I),\ \ i = 0, 1, 2, 3 \tag{5}$$

where d_i and G_i denote the dilated rates and groups of each branch, respectively. Among them, $d_i = [1, 3, 5, 7]$ and $G_i = [1, 4, 8, 10]$. Finally, the output can be produced by concatenating them, which is defined as,

$$O = \mathrm{Cat}(F_0, F_1, F_2, F_3) \tag{6}$$

where Cat means a concatenation operation. With the aid of group convolutions and dilated convolutions, the multiscale representations can be obtained efficiently.

3.4. Ground truth generation

Similar to the most previous crowd counting methods [8,10,39], we employ the Gaussian kernel G_σ blur dot annotations to generate the ground truth map. Assuming a head annotation is at pixel x_i, it can be represented as $\delta(x - x_i)$. Then the ground truth map is obtained by convolving $\delta(x - x_i)$. The process can be formulated as,

$$M_{gt} = \sum_{i=1}^{H} \delta(x - x_i) * G_\sigma, \tag{7}$$

where H denotes the number of labelled heads in an image, and $*$ represents a convolution operation. σ is Gaussian kernel size, and it is set to a constant of 15 in all experiments.

3.5. Loss function

We adopt the Euclidean loss to evaluate the difference between the prediction and target map at a pixel level. It is formulated as,

$$L_{count} = \frac{1}{T}\sum_{t=1}^{T} \left\| X_{est}(I_t, \theta) - X_{gt}(I_t) \right\|_2^2, \tag{8}$$

where $X_{est}(I_t, \theta)$ and $X_{gt}(I_t)$ denote the estimated and ground truth map, respectively. T is the total pixels in the density map. I_k and θ represent the kth input and a set of parameters to be trained.

4. Experiments

4.1. Datasets

Five benchmark crowd counting datasets are utilized to conduct the experiments. The specific details of the datasets are listed in Table 2.

ShanghaiTech. The ShanghaiTech dataset [10] consists of two parts, *i.e.,* Part A and Part B. Part A is randomly downloaded from the website and has a higher crowd density than Part B, whereas Part B is taken from the real-scene streets and has more obvious non-uniform distribution and scale variation than Part A.

UCF_CC_50. The UCF_CC_50 dataset [34] has extremely high pedestrian density and the limited sample size. The two serious

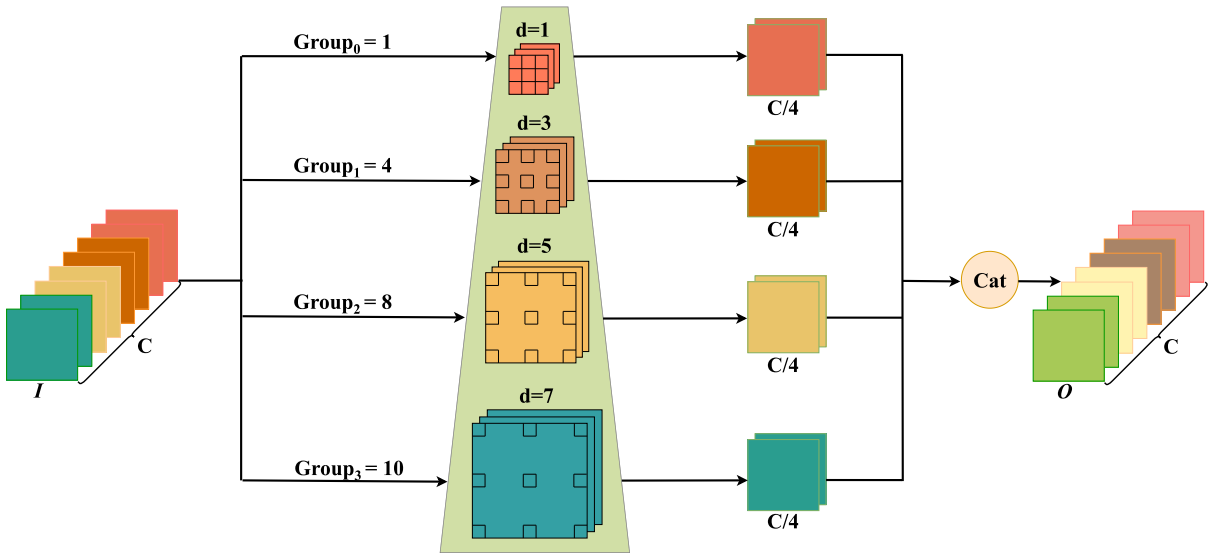

Fig. 6. Architecture of the proposed EPF module. First, it divides the input into four groups. Then, convolution layers with different dilated rates are adopted to obtain features with different scales. Last, a concatenation operation is utilized to generate the output.

troubles make most of the current methods perform poorly on the dataset. Due to the small number of images, 5-fold cross-validation is performed to train and test the proposed model.

UCF-QNRF. The UCF-QNRF dataset [35] is one of the most challenging crowd counting datasets because of the scene diversity and large scale variation. The images are randomly collected from the Internet, and the average resolution is higher than other datasets.

WorldExpo'10. The WorldExpo'10 dataset [36] is captured from 2010 Shanghai World Expo and is composed of video sequences of five scenes. Furthermore, it gives the region of interest (ROI) for each scene.

NWPU-Crowd. The NWPU-Crowd dataset [37] is the largest crowd counting dataset in terms of sample size and resolution till now. It is taken from the self-shooting and Internet and is divided into five levels based on the scene density. In addition, it provides both point-level and bounding box level annotations.

CARPK [38] is specified for vehicle counting, and it is taken from four diverse parking lots in a drone-view.

4.2. Implementation details

For data augmentation at the training stage, randomly cropped with a size of 576 × 768 and horizontally flipped are conducted on all images. The batch size is set to 8 for all datasets. The maximum number of epochs is set to 3,000. Adam algorithm [40] with a learning rate of 1e-4 and a decay rate of 0.995 is used to optimize the model. All experiments are implemented by the PyTorch toolbox. An RTX 3090Ti NVIDIA GPU is employed to train the model. Four different types of GPUs, *i.e.*, RTX 3090Ti, RTX 3090, RTX 2080 super, and GTX 1080Ti, are utilized for model efficiency test.

4.3. Evaluation protocols

Consistent with the previous methods [9,10], we utilize the mean absolute error (MAE) and root mean square error (RMSE) to evaluate the counting methods in accuracy and stability. They are defined as,

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|Est_i - Gt_i|, \tag{9}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\|Est_i - Gt_i\|_2^2}, \tag{10}$$

where N represents the number of test samples, Est_i and Gt_i are the estimated counts and real counts of ith image, respectively. Lower MAE and RMSE indicate better counting accuracy and stability of the model.

To evaluate the efficiency of the models, we adopt the indicator of FLOPs [17,19], inferring time and FPS (frames per second).

4.4. Comparison and analysis

We compare the proposed GAPNet with other mainstream methods in counting performance and efficiency. The comparison results are shown in Table 3, Table 4 and Table 5, respectively.

4.4.1. Crowd counting comparison

The comparison of counting results is reported in Table 3. We divide all methods into two groups based on the number of parameters (separated by horizontal line). Group 1 represents the dense networks, whereas the Group 2 denotes the lightweight networks. Overall, the results of the first group are superior to the second group because they have more complex models and more parameters.

On Part A, the GAPNet gets the first place in both MAE and RMSE, and has a big margin with other lightweight models. Specifically, compared with the second-best method PCCNet [19], the proposed model improves the MAE and RMSE by 8.7% and 11.0%, respectively. In addition, the GAPNet is also competitive compared with the first group. Compared with CSRNet [25], which also aims to deal with scale variation, the proposed method reduces MAE and RMSE by 1.6% and 4%, respectively. In addition to the improved counting performance, the GAPNet reduces the parameters by 5.7 times compared to CSRNet, indicating that the proposed method is more suitable to be deployed to actual scenes. On the relatively sparse Part B, the GAPNet also achieves satisfactory results, with scores of 9.8 and 15.2 in MAE and RMSE.

On the UCF_CC_50 dataset, the counting performance of the proposed GAPNet outperforms again all the other competitors in

Table 3
Objective comparison results on the crowd benchmark datasets. For the first group, the best performance is marked in red, and the second-best performance is marked in blue. For the second group, the best results are highlighted in **bold**, and the sub-best results are highlighted in underline.

Methods	Part A		Part B		UCF_CC_50		UCF-QNRF		WorldExpo'10						NWPU		Params (M)
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	S1	S2	S3	S4	S5	Avg	MAE	RMSE	
CAN [13]	62.3	100.0	7.8	12.2	212.2	243.7	107.0	183.0	2.9	12.0	10.0	7.9	4.3	7.4	106.3	386.5	18.10
BL [14]	61.5	103.2	7.5	12.6	229.3	308.2	87.7	158.1	–	–	–	–	–	–	105.4	454.2	21.50
ASNet [24]	57.8	90.0	–	–	174.8	251.6	91.6	159.7	2.2	10.1	8.9	7.1	4.8	6.6	–	–	30.39
NoiseCC [41]	61.9	99.6	7.4	11.3	–	–	85.8	150.6	1.6	8.8	10.8	10.4	2.5	6.8	96.9	534.2	20.02
LSC-CNN [42]	66.5	101.8	7.7	12.7	225.6	302.7	120.5	218.2	2.9	11.3	9.4	12.3	4.3	8.0	–	–	42.23
SFCN [43]	64.8	107.5	7.6	13.0	214.2	318.2	102.0	171.4	1.8	17.5	11.1	13.5	3.0	9.4	105.7	424.1	38.60
UOT [44]	58.1	95.9	6.5	10.2	–	–	83.3	142.3	–	–	–	–	–	–	87.8	387.5	21.50
DKPNet [27]	55.6	91.0	6.6	10.9	–	–	81.4	147.2	–	–	–	–	–	–	74.5	327.4	30.63
GL [45]	61.3	95.4	7.3	11.1	–	–	84.3	147.5	–	–	–	–	–	–	79.3	346.1	21.51
CSRNet [25]	68.2	115.0	10.6	16.0	266.1	397.5	135.4	207.4	2.9	11.5	8.6	16.6	3.4	8.6	121.3	387.8	16.26
UEPNet [16]	54.6	91.2	6.4	10.9	165.2	275.9	81.1	131.7	1.6	9.8	10.3	8.6	4.1	6.9	–	–	26.12
P2PNet [7]	52.7	85.1	6.3	9.9	172.7	256.1	85.3	154.5	–	–	–	–	–	–	77.4	362.0	18.34
STNet [20]	52.9	83.6	6.3	10.3	162.0	230.4	87.9	166.4	–	–	–	–	–	–	–	–	15.56
MCNN [10]	110.2	173.2	26.4	41.3	377.6	509.1	277.0	426.0	3.4	20.6	12.9	13.0	8.1	11.6	232.5	714.6	0.13
PCCNet [19]	73.5	124.0	11.0	19.0	240.0	315.5	148.7	247.3	1.9	18.3	10.5	13.4	3.4	9.5	–	–	0.55
SANet [29]	75.3	122.2	10.5	17.9	258.4	334.9	152.6	547.0	2.6	13.2	9.0	13.3	3.0	8.2	190.6	**491.4**	0.91
LCNet [31]	93.3	149.0	15.3	25.2	326.7	430.6	–	–	–	–	–	–	–	–	–	–	0.86
CCNN [32]	88.1	141.7	14.9	22.1	–	–	–	–	3.8	20.5	8.8	8.8	7.7	9.9	–	–	0.073
1/4SAN+SKT [18]	78.0	126.6	11.9	19.8	–		157.5	257.7	3.4	16.1	15.8	15.4	4.9	11.1	–	–	0.058
MobileCount [17]	89.4	146.0	**9.0**	15.4	284.8	392.8	131.1	222.6	–	–	–	–	–	11.1	–	–	3.40
GAPNet (Ours)	**67.1**	**110.4**	9.8	**15.2**	**202.8**	**246.9**	**118.5**	**217.2**	**1.5**	**11.5**	**8.0**	**7.0**	**2.5**	**6.1**	**174.1**	514.7	2.85

Table 4
Comparison results of different models in inferring time and FPS on Shanghai Part A using different GPUs. (The input size is set to 576 × 768.).

Methods	Params (M)	FLOPs	RTX 3090Ti		RTX 3090		RTX 2080 super		GTX 1080Ti	
			Time (ms)	FPS	Time (ms)	FPS	Time (ms)	FPS	Time (ms)	FPS
MCNN [10]	0.13	11.9	4.8	210.7	4.2	235.5	8.6	116.0	9.3	107.1
CSRNet [25]	16.26	182.7	16.6	60.4	15.1	66.4	34.4	29.0	43.0	23.3
CAN [13]	18.10	193.7	18.9	52.8	18.3	54.6	35.6	25.3	49.8	20.1
BL [14]	21.50	182.2	16.3	61.3	15.7	63.4	30.1	33.2	31.6	31.7
SASNet [15]	38.90	393.2	43.1	23.2	45.3	22.1	90.2	11.1	100.6	9.9
GAPNet (Ours)	2.85	3.29	4.0	252.1	7.7	130.1	9.7	102.8	10.4	99.6

the second group, and it is quite comparable along the methods in the first group. Compared with the SFCN [43], the GAPNet improves the MAE and RMSE by 5.3% and 22.4%, respectively. In fact, SFCN has 13.5 times more parameters than the GAPNet. This proves that the proposed method not only ensures the lightweight of the model, but also guarantees the counting accuracy.

On the UCF-QNRF dataset, the GAPNet scores 118.5 and 217.2 in MAE and RMSE, respectively, both outperforming the methods in the second group. Compared with the second-best method MobileCount [17], it achieves an improvement by 9.6%, 2.4%, and 16.2% in terms of MAE, RMSE and parameter size. Additionally, because UCF-QNRF is characterized by large scale variation, the experimental results prove that GAPNet can cope with this challenge well.

On the WorldExpo'10 dataset, one can see that the proposed methods performs best in the scenes of 1, 3, 4, 5 and the average score among both Group 1 and Group 2 in terms of MAE. The only special case is the scene 2 in which the winner is the NoiseCC [41]. However, the GAPNet improves the average MAE value by 10.3% compared with NoiseCC. Also, it can be seen that the ASNet [24] performs best in MAE within Group 1. Compared with ASNet, the GAPNet has improvements of 7.6% in MAE and 90.6% in Params.

On the NWPU-Crowd dataset, it can be observed that the GAPNet scores 174.1 and 514.7 in MAE and RMSE, respectively. Compared with lightweight networks in Group 2, it performs the best in MAE and ranks only second to SANet in RMSE. The reason can be attributed that we only use MSE loss function for network training, but SANet employs MSE and local pattern consistent loss to enhance the local correlation of density map. Still, it has a large improvement room compared with the methods in Group 1. Because the methods in Group 1 has larger number of parameters, which are beneficial to regress a high-quality density map compared with the lightweight methods.

Table 5
Objective experimental results on the CARPK dataset.

Methods	MAE	RMSE	Params(M)
LCDNet [46]	13.1	–	0.21
DroneNet [48]	9.0	–	0.6
LMSFFNet [47]	7.1	9.0	4.58
MoblieCount [17]	17.5	23.9	3.40
GAPNet (Ours)	8.9	13.7	2.85

Some subjective results are illuminated in Fig. 7. 'GT' is the ground truth. 'Est' represents the number of people predicted by the network, which is obtained by summing the pixels on the density map. It demonstrates that the estimated map counts are closely approximate to the ground truth values. The training curve is shown in Fig. 8.

4.4.2. Vehicle counting comparison

To explore the generalization of the proposed GAPNet, we compared it with several lightweight models (LCDNet [46], LMSFFNet [47], MoblieCount [17], DroneNet [48]) on CARPK dataset. The comparison results are listed in Table 5. Among the competitors, LCDNet [46], LMSFFNet [47], and DroneNet [48] are specified for vehicle counting. One can see that the proposed GAPNet is slightly inferior to the LMSFFNet, but it outperforms the LCDNet and DroneNet in terms of accuracy. Specially, the proposed GAPNet is superior to LMSFFNet in terms of Params. Furthermore, compared with the MobileCount, it reduces the MAE and RMSE by 49.1% and 42.7% respectively, and it has fewer parameters.

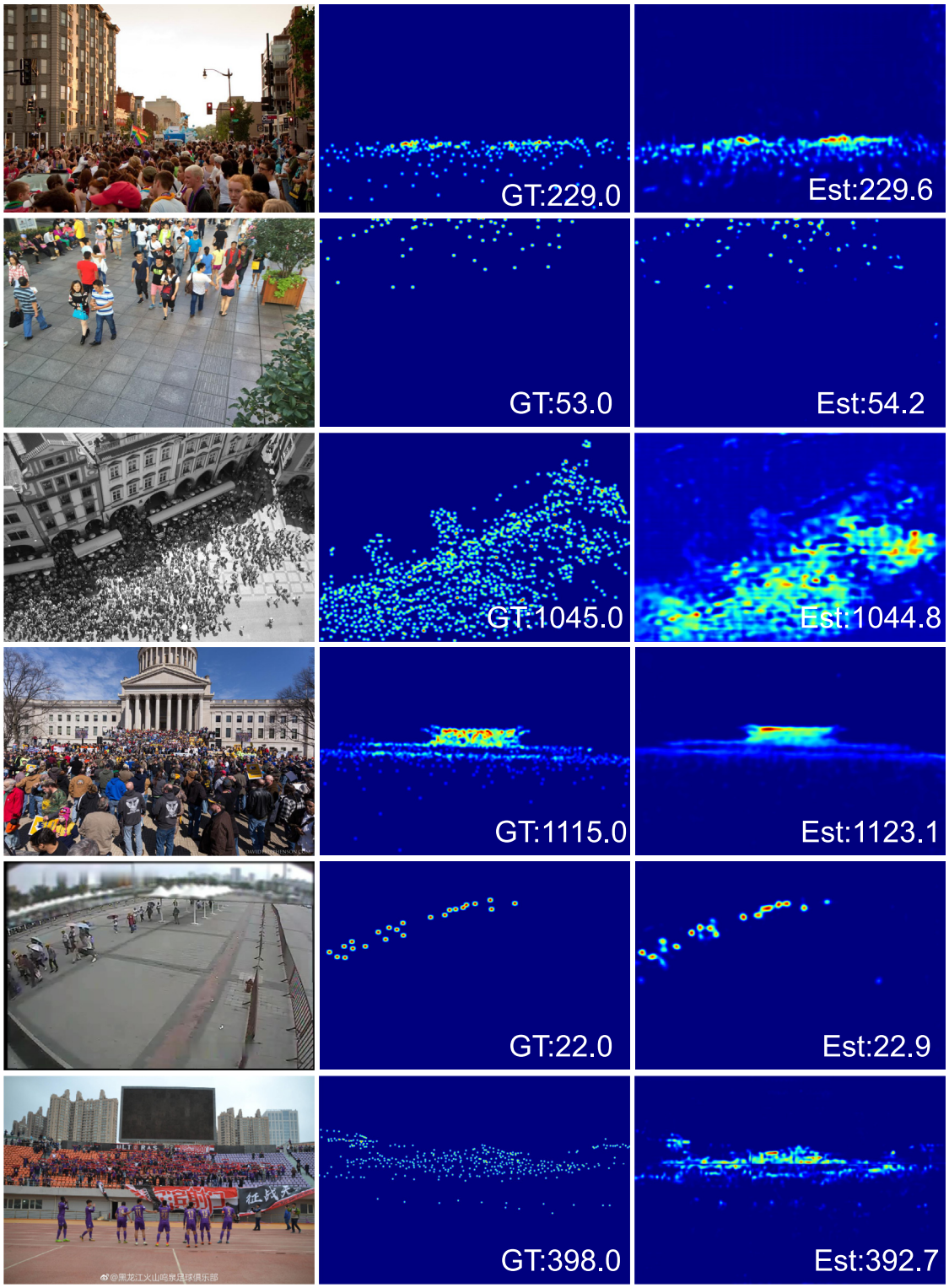

Fig. 7. Subjective Comparisons of different methods on the datasets. From top to bottom, it depicts the input, ground truth and estimated density maps of Shanghai Part A, Shanghai Part B, UCF_CC_50, UCF-QNRF, World Expo'10 and NWPU-Crowd datasets.

4.4.3. Efficiency comparison

To test the efficiency of the proposed method, we conduct a series of comparative experiments on Shanghai Part A dataset using four different GPUs (RTX 3090Ti, RTX 3090, RTX 2080 super and GTX 1080 Ti). The input size is set to 576 × 768.

Three evaluation indicators, namely FLOPs, inferring time and frame per second (FPS), are used to evaluate the efficiency of different models. The comparative results are listed in Table 4. It proves that the complex models, *e.g.,* (CSRNet [25], CAN [13], BL [14] and SASNet [15]) have higher FLOPs, longer reasoning time and lower FPS, which indicates that the networks are inefficient and are difficult to be deployed in real-world applications. In contrast, MCNN [10] has the most light model with a parameter of 0.13M but has the worst counting performance. Compared with

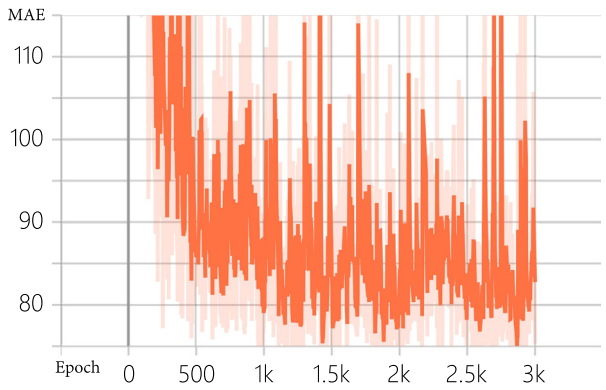

Fig. 8. The MAE curve during the training process on the Shanghai Part A dataset.

MCNN, the GAPNet achieves better result in terms of inferring time and FPS. On an RTX 3090Ti, the GAPNet gets the fastest inferring speed and highest FPS with 4.0 and 252.1, outperforming the other methods.

4.5. Ablation study

To explore the effectiveness of the proposed ZCA module and EPF module, we conduct a set of ablation studies on Shanghai Part A dataset. The results are reported in Table 6. The baseline represents the model only with encoder and decoder. The symbols of '+' and '||' denote the connection mode of series and parallel, respectively.

One can observe that the baseline scores 72.8 and 120.8 in MAE and RMSE. When the ZCA module is deployed into the baseline, there are no growth of parameters and FLOPs, but the MAE and RMSE improve by 3.0% and 7.7%, respectively. Moreover, compared with the classical channel attention SE module [49], the ZCA module can achieve the same counting performance. When the EPF module is adopted in the baseline, the MAE and RMSE are improved by 4.8% and 4.7% with only a 0.06M parameter increase. As can be observed from the different connection modes between the ZCA and EPF modules, the series mode performs better than parallel mode in accuracy. In the two modes of series, the mode of 'Baseline+ZCA+EPF' is better than 'Baseline+EPF+ZCA'.

The visual results of different configurations on Shanghai Part A dataset is depicted in Fig. 9. From top left to bottom right are the original input, ground truth and the estimated density maps by seven different configurations. One can observe that the mode 5, i.e., 'Baseline+EPF||ZCA' has a poor performance in generating the density map and the estimated counts are far away from the ground truth. Compared with other configuration modes, the estimated density map and the counts of the GAPNet are closet to the ground truth.

5. Conclusion and future work

In this paper, we propose a lightweight GAPNet to solve the scale variation in crowd counting. The GAPNet consists of an encoder, a ZCA module, an EPF module and a decoder. Specifically, the encoder can discard the redundant features while extracting the basic feature representations. The ZCA module is capable to select a crowd region while consuming zero parameters. The EPF module builds a pyramid structure to extract multiscale features by deploying group convolutions and dilated convolutions. The

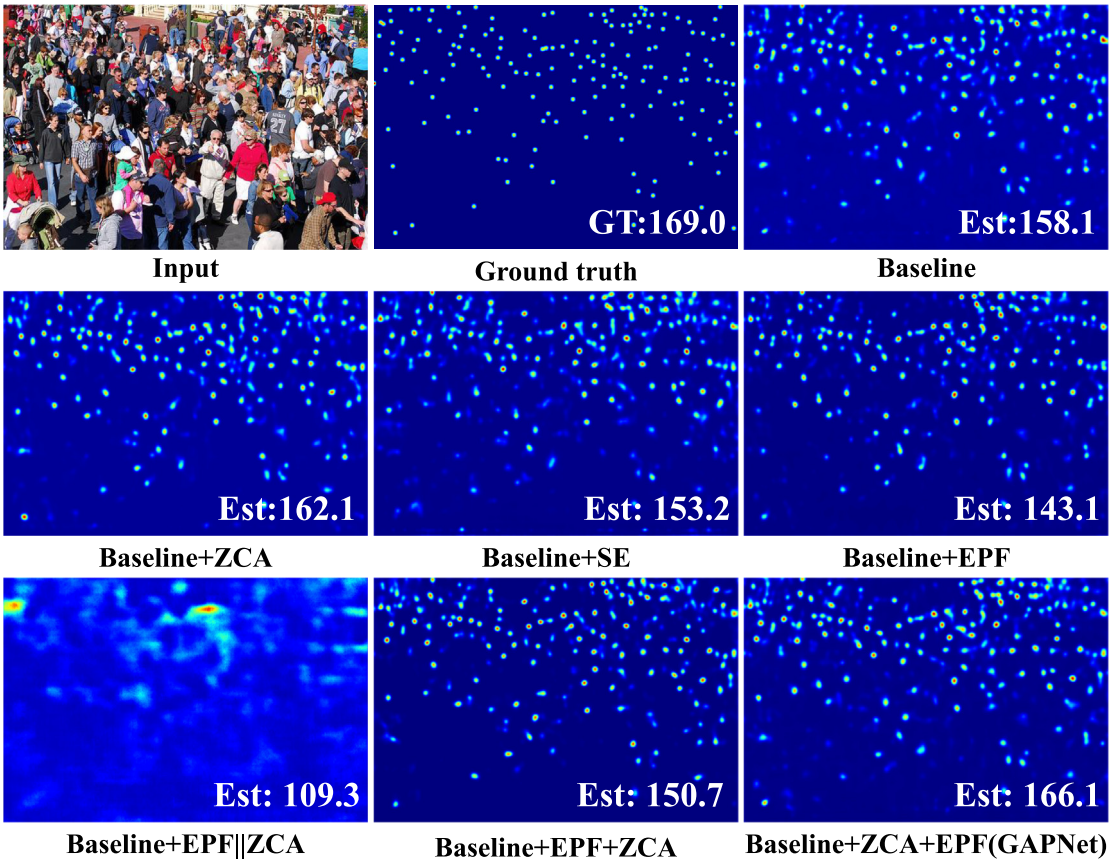

Fig. 9. Subjective comparison of different methods on Shanghai Part A dataset.

Table 6
Ablation studies on the critical modules in the GAPNet.

Methods	Params	FLOPs	MAE	RMSE
Baseline	2.79	3.1936	72.8	120.8
Baseline+ZCA	2.79	3.1936	70.6	111.5
Baseline+SE	2.80	3.1936	70.5	111.8
Baseline+EPF	2.85	3.2933	69.3	115.1
Baseline+EPF‖ZCA	2.85	3.2933	75.5	120.3
Baseline+EPF+ZCA	2.85	3.2932	68.6	113.8
Baseline+ZCA+EPF	2.85	3.2933	**67.1**	**110.4**

two modules jointly solve the scale problem effectively and efficiently. Finally, we stack a set of transposed convolution blocks as a decoder to generate a high-quality density map. Comparison results on five crowd counting datasets verify the superiority of the GAPNet in counting accuracy and efficiency. In future work, we intend to further improve the proposed lightweight models by combining the latest computational compress technologies, and deploy them into edge devices.

CRediT authorship contribution statement

Xiangyu Guo: Conceptualization, Validation, Data curation, Investigation. **Kai Song:** Methodology, Visualization, Data analysis. **Mingliang Gao:** Investigation, Validation. **Wenzhe Zhai:** Visualization, Software. **Qilei Li:** Software. **Gwanggil Jeon:** Reviewing and editing.

Declaration of competing interest

The authors declare no conflict of interest.

Data availability

No data was used for the research described in the article.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (Nos. 61601266 and 61801272).

References

[1] R. Sundararaman, C. De Almeida Braga, E. Marchand, J. Pettré, Tracking pedestrian heads in dense crowd, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2021, pp. 3864–3874, http://dx.doi.org/10.1109/CVPR46437.2021.00386.

[2] G. Gao, J. Gao, Q. Liu, Q. Wang, Y. Wang, CNN-based density estimation and crowd counting: A survey, 2020, ArXiv arXiv:2003.12783.

[3] G. Castellano, C. Mencar, G. Sette, F.S. Troccoli, G. Vessio, Crowd flow detection from drones with fully convolutional networks and clustering, in: 2022 International Joint Conference on Neural Networks, IJCNN, 2022, pp. 1–8, http://dx.doi.org/10.1109/IJCNN55064.2022.9891954.

[4] Z. Fan, H. Zhang, Z. Zhang, G. Lu, Y. Zhang, Y. Wang, A survey of crowd counting and density estimation based on convolutional neural network, Neurocomputing 472 (2022) 224–251, http://dx.doi.org/10.1016/j.neucom.2021.02.103.

[5] I.S. Topkaya, H. Erdogan, F.M. Porikli, Counting people by clustering person detector outputs, in: 2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS, 2014, pp. 313–318, http://dx.doi.org/10.1109/AVSS.2014.6918687.

[6] V. Lempitsky, A. Zisserman, Learning to count objects in images, in: Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS), 2010, pp. 1324–1332.

[7] Q. Song, C. Wang, Z. Jiang, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, Y. Wu, Rethinking counting and localization in crowds: A purely point-based framework, in: 2021 IEEE/CVF International Conference on Computer Vision, ICCV, 2021, pp. 3345–3354, http://dx.doi.org/10.1109/ICCV48922.2021.00335.

[8] W. Zhai, Q. Li, Y. Zhou, X. Li, J. Pan, G. Zou, M. Gao, DA2net: a dual attention-aware network for robust crowd counting, in: Multimedia Systems, 2022, pp. 1432–1882, http://dx.doi.org/10.1007/s00530-021-00877-4.

[9] X. Guo, M. Gao, W. Zhai, Q. Li, J. Pan, G. Zou, Multiscale aggregation network via smooth inverse map for crowd counting, in: Multimedia Tools and Applications, 2022, http://dx.doi.org/10.1007/s11042-022-13664-8.

[10] Y. Zhang, D. Zhou, S. Chen, S. Gao, Y. Ma, Single-image crowd counting via multi-column convolutional neural network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 589–597, http://dx.doi.org/10.1109/CVPR.2016.70.

[11] D.B. Sam, R.V. Babu, Top-down feedback for crowd counting convolutional neural network, in: Proceedings of the AAAI Conference on Artificial Intelligence, AAAI, 2018, pp. 7323–7330, http://dx.doi.org/10.1609/aaai.v32i1.12290.

[12] R.R. Varior, B. Shuai, J. Tighe, D. Modolo, Scale-aware attention network for crowd counting, 2019, ArXiv arXiv:1901.06026.

[13] W. Liu, M. Salzmann, P. Fua, Context-aware crowd counting, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2019, pp. 5094–5103, http://dx.doi.org/10.1109/CVPR.2019.00524.

[14] Z. Ma, X. Wei, X. Hong, Y. Gong, Bayesian loss for crowd count estimation with point supervision, in: Proceedings of the International Conference on Computer Vision, ICCV, 2019, pp. 6141–6150, http://dx.doi.org/10.1109/ICCV.2019.00624.

[15] Q. Song, C. Wang, Y. Wang, Y. Tai, C. Wang, J. Li, J. Wu, J. Ma, To choose or to fuse? Scale selection for crowd counting, in: AAAI, 2021, pp. 2576–2583, http://dx.doi.org/10.1609/aaai.v35i3.16360.

[16] C. Wang, Q. Song, B. Zhang, Y. Wang, Y. Tai, X. Hu, C. Wang, J. Li, J. Ma, Y. Wu, Uniformity in heterogeneity: Diving deep into count interval partition for crowd counting, in: 2021 IEEE/CVF International Conference on Computer Vision, ICCV, 2021, pp. 3214–3222, http://dx.doi.org/10.1109/ICCV48922.2021.00322.

[17] P. Wang, C. Gao, Y. Wang, H. Li, Y. Gao, MobileCount: An efficient encoder-decoder framework for real-time crowd counting, Neurocomputing 407 (2020) 292–299, http://dx.doi.org/10.1016/j.neucom.2020.05.056.

[18] L. Liu, J. Chen, H. Wu, T. Chen, G. Li, L. Lin, Efficient crowd counting via structured knowledge transfer, in: Proceedings of the ACM International Conference on Multimedia (ACM MM), 2020, pp. 2645–2654, http://dx.doi.org/10.1145/3394171.3413938.

[19] J. Gao, Q. Wang, X. Li, PCC net: Perspective crowd counting via spatial convolutional network, IEEE Trans. Circuits Syst. Video Technol. 30 (2020) 3486–3498, http://dx.doi.org/10.1109/TCSVT.2019.2919139.

[20] M. Wang, H. Cai, X. Han, J. Zhou, M. Gong, STNet: Scale tree network with multi-level auxiliator for crowd counting, in: IEEE Transactions on Multimedia, 2022, http://dx.doi.org/10.1109/TMM.2022.3142398, arXiv:2012.10189.

[21] X. Guo, M. Gao, W. Zhai, J. Shang, Q. Li, Spatial-frequency attention network for crowd counting., Big Data 10 (5) (2022) 453–465, http://dx.doi.org/10.1089/big.2022.0039.

[22] D. Guo, K. Li, Z. Zha, M. Wang, DADNet: Dilated-attention-deformable ConvNet for crowd counting, in: Proceedings of the ACM International Conference on Multimedia (ACM MM), 2019, pp. 1823–1832, http://dx.doi.org/10.1145/3343031.3350881.

[23] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, C. Xu, GhostNet: More features from cheap operations, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2020, pp. 1577–1586, http://dx.doi.org/10.1109/cvpr42600.2020.00165.

[24] X. Jiang, L. Zhang, M. Xu, T. Zhang, P. Lv, B. Zhou, X. Yang, Y. Pang, Attention scaling for crowd counting, in: CVPR, 2020, pp. 4705–4714, http://dx.doi.org/10.1109/cvpr42600.2020.00476.

[25] Y. Li, X. Zhang, D. Chen, Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2018, pp. 1091–1100, http://dx.doi.org/10.1109/CVPR.2018.00120.

[26] V.A. Sindagi, V.M. Patel, HA-CCN: Hierarchical attention-based crowd counting network, IEEE Trans. Image Process. 29 (2020) 323–335, http://dx.doi.org/10.1109/TIP.2019.2928634.

[27] B. Chen, Z. Yan, K. Li, P. Li, B. Wang, W. Zuo, L. Zhang, Variational attention: Propagating domain-specific knowledge for multi-domain learning in crowd counting, in: 2021 IEEE/CVF International Conference on Computer Vision, ICCV, 2021, pp. 16045–16055, http://dx.doi.org/10.1109/ICCV48922.2021.01576.

[28] Q. Wang, T. Breckon, Crowd counting via segmentation guided attention networks and curriculum loss, IEEE Trans. Intell. Transp. Syst. 23 (2022) 15233–15243, http://dx.doi.org/10.1109/TITS.2021.3138896.

[29] X. Cao, Z. Wang, Y. Zhao, F. Su, Scale aggregation network for accurate and efficient crowd counting, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 734–750, http://dx.doi.org/10.1007/978-3-030-01228-1_45.

[30] M. Sandler, A.G. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, MobileNetV2: Inverted residuals and linear bottlenecks, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510–4520, http://dx.doi.org/10.1109/CVPR.2018.00474.
[31] X. Ma, S. Du, Y. Liu, A lightweight neural network for crowd analysis of images with congested scenes, in: 2019 IEEE International Conference on Image Processing, ICIP, 2019, pp. 979–983, http://dx.doi.org/10.1109/ICIP.2019.8803062.
[32] X. Shi, X. Li, C. Wu, S. Kong, J.-S. Yang, L. He, A real-time deep network for crowd counting, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, 2020, pp. 2328–2332, http://dx.doi.org/10.1109/ICASSP40776.2020.9053780.
[33] L. Yang, R.-Y. Zhang, L. Li, X. Xie, Simam: A simple, parameter-free attention module for convolutional neural networks, in: ICML, 2021, pp. 11863–11874.
[34] H. Idrees, I. Saleemi, C. Seibert, M. Shah, Multi-source multi-scale counting in extremely dense crowd images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2013, pp. 2547–2554, http://dx.doi.org/10.1109/CVPR.2013.329.
[35] H. Idrees, M. Tayyab, K. Athrey, D. Zhang, S. Al-Maadeed, N. Rajpoot, M. Shah, Composition loss for counting, density map estimation and localization in dense crowds, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 532–546, http://dx.doi.org/10.1007/978-3-030-01216-8_33.
[36] C. Zhang, H. Li, X. Wang, X. Yang, Cross-scene crowd counting via deep convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2015, pp. 833–841, http://dx.doi.org/10.1109/CVPR.2015.7298684.
[37] Q. Wang, J. Gao, W. Lin, X. Li, NWPU-crowd: A large-scale benchmark for crowd counting and localization, IEEE Trans. Pattern Anal. Mach. Intell. 43 (2021) 2141–2149, http://dx.doi.org/10.1109/TPAMI.2020.3013269.
[38] M.-R. Hsieh, Y.-L. Lin, W.H. Hsu, Drone-based object counting by Spatially Regularized Regional proposal network, in: Proceedings of the International Conference on Computer Vision, ICCV, 2017, pp. 4165–4173.
[39] M.-H. Guo, T. Xu, J. Liu, Z.-N. Liu, P.-T. Jiang, T.-J. Mu, S.-H. Zhang, R.R. Martin, M.-M. Cheng, S. Hu, Attention mechanisms in computer vision: A survey, 2022, ArXiv arXiv:2111.07624.
[40] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Proceedings of the International Conference on Learning Representations, ICLR, 2015.
[41] J. Wan, A.B. Chan, Modeling noisy annotations for crowd counting, in: Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS), 2020, pp. 3386–3396.
[42] D.B. Sam, S.V. Peri, M.N. Sundararaman, A. Kamath, R.V. Babu, Locate, size, and count: Accurately resolving people in dense crowds via detection, IEEE Trans. Pattern Anal. Mach. Intell. 43 (2021) 2739–2751, http://dx.doi.org/10.1109/tpami.2020.2974830.
[43] W. Qi, J. Gao, L. Wei, Y. Yuan, Pixel-wise crowd understanding via synthetic data, Int. J. Comput. Vis. 129 (2021) 225–245, http://dx.doi.org/10.1007/s11263-020-01365-4.
[44] Z. Ma, X. Wei, X. Hong, H. Lin, Y. Qiu, Y. Gong, Learning to count via unbalanced optimal transport, in: AAAI, 2021, pp. 2319–2327, http://dx.doi.org/10.1609/aaai.v35i3.16332.
[45] J. Wan, Z. Liu, A.B. Chan, A generalized loss function for crowd counting and localization, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2021, pp. 1974–1983, http://dx.doi.org/10.1109/CVPR46437.2021.00201.
[46] M.A. Khan, H. Menouar, R. Hamila, Lcdnet: a lightweight crowd density estimation model for real-time video surveillance, J. Real-Time Image Process. 20 (2023).
[47] J. Yi, Z. Shen, F. Chen, Y. Zhao, S. Xiao, W. Zhou, A lightweight multiscale feature fusion network for remote sensing object counting, IEEE Trans. Geosci. Remote Sens. 61 (2023) 1–13, http://dx.doi.org/10.1109/TGRS.2023.3238185.
[48] M.A. Khan, H. Menouar, R. Hamila, DroneNet: Crowd density estimation using self-ONNs for drones, in: 2023 IEEE 20th Consumer Communications & Networking Conference (CCNC), 2022, pp. 455–460.
[49] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2018, pp. 7132–7141, http://dx.doi.org/10.1109/TPAMI.2019.2913372.

Xiangyu Guo is pursuing his M.S. degree at the School of Electrical and Electronic Engineering, Shandong University of Technology, Zibo, China. His research interests include image processing, deep learning, and crowd analysis.

Kai Song received his M.S. degree from the China University Of Petroleum in 2010. Now, he is a senior engineer in Information Service Center, Zhengzhou, China. His research interests include signal processing and deep learning.

Mingliang Gao received his Ph.D. in communication and information systems from Sichuan University, Chengdu, China, in 2013. He is currently an associate professor at the School of Electrical and Electronic Engineering, Shandong University of Technology, Zibo, China. His main research interests include computer vision and deep learning.

Wenzhe Zhai is pursuing his M.S. degree at the School of Electrical and Electronic Engineering, Shandong University of Technology, Zibo, China. His research interests include image processing and pattern recognition.

Qilei Li is currently a Ph.D. student with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London, United Kingdom. He received the M.S. degree in signal and information processing from Sichuan University. His research interests are computer vision and deep learning.

Gwanggil Jeon received the B.S., M.S., and Ph.D. (summa cum laude) degrees from the Department of Electronics and Computer Engineering, Hanyang University, Seoul, Korea, in 2003, 2005, and 2008, respectively.