Efficient Crowd Counting via Dual Knowledge Distillation

Rui Wang, Yixue Hao, *Member, IEEE*, Long Hu, Xianzhi Li, Min Chen, *Fellow, IEEE*, Yiming Miao, *Member, IEEE*, and Iztok Humar, *Senior Member, IEEE*

***Abstract*— Most researchers focus on designing accurate crowd counting models with heavy parameters and computations but ignore the resource burden during the model deployment. A real-world scenario demands an efficient counting model with low-latency and high-performance. Knowledge distillation provides an elegant way to transfer knowledge from a complicated teacher model to a compact student model while maintaining accuracy. However, the student model receives the wrong guidance with the supervision of the teacher model due to the inaccurate information understood by the teacher in some cases. In this paper, we propose a dual-knowledge distillation (DKD) framework, which aims to reduce the side effects of the teacher model and transfer hierarchical knowledge to obtain a more efficient counting model. First, the student model is initialized with global information transferred by the teacher model via adaptive perspectives. Then, the self-knowledge distillation forces the student model to learn the knowledge by itself, based on intermediate feature maps and target map. Specifically, the optimal transport distance is utilized to measure the difference of feature maps between the teacher and the student to perform the distribution alignment of the counting area. Extensive experiments are conducted on four challenging datasets, demonstrating the superiority of DKD. When there are only approximately 6% of the parameters and computations from the original models, the student model achieves a faster and more accurate counting performance as the teacher model even surpasses it.**

***Index Terms*— Crowd counting, knowledge transfer, self-knowledge distillation, optimal transport distance.**

Manuscript received 9 November 2022; revised 6 August 2023 and 29 October 2023; accepted 5 December 2023. Date of publication 21 December 2023; date of current version 4 January 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2023YFB4503400; in part by the National Natural Science Foundation of China (NSFC) under Grant 62176101, Grant 62276109, Grant 62322205, Grant 62202410, and Grant 62202182; and in part by the Shenzhen Science and Technology Program under Grant JCYJ20220530143808019. The work of Yiming Miao was supported in part by the Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS). The work of Iztok Humar was supported the Slovenian Research and Innovation Agency through the Decentralized solutions for the Digitalization of Industry and Smart Cities and Communities under Research Program P2-0425. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jun Cheng. *(Corresponding author: Min Chen.)*

Rui Wang, Yixue Hao, Long Hu, and Xianzhi Li are with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: ruiwang2020@hust.edu.cn; yixuehao@hust.edu.cn; hulong@hust.edu.cn; xzli@hust.edu.cn).

Min Chen is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510640, China, and also with the Pazhou Laboratory, Guangzhou 510330, China (e-mail: minchen@ieee.org).

Yiming Miao is with the Shenzhen Institute of Artiffcial Intelligence and Robotics for Society (AIRS), School of Data Science, The Chinese University of Hong Kong, Shenzhen (CUHK-Shenzhen), Guangdong 518172, P.R. China (e-mail: yimingmiao@ieee.org).

Iztok Humar is with the Faculty of Electrical Engineering, University of Ljubljana, 1000 Ljubljana, Slovenia (e-mail: iztok.humar@fe.uni-lj.si).

Digital Object Identifier 10.1109/TIP.2023.3343609

I. Introduction

CROWD counting, a critical computer-vision task that aims to estimate the number of people in monitoring scenarios, has achieved remarkable progress [1], [2], [3], [4], [5]. Recently, researchers have paid more attention to the counting application, trying to efficiently obtain crowd estimation in the real-time system with less computation consumption.

Convolutional Neural Networks (CNNs) are widely used in the detection, regression, density-map, and point-supervision counting estimation methods. Various networks, such as multi-column [6], [7], [8], dilated [9], [10], deconvolutional [11], and pre-trained CNNs, such as VGG and ResNet, are used as the backbones of the counting network. Further, transformer structures are applied to the counting task, achieving better results than the CNN [12], [13]. Benefiting from the feature extraction capability in complex scenes and the ingenious design for optimization objectives by these models, the counting accuracy has constantly improved. To further promote the deployment of counting models on the edge server and terminals, some lightweight models have been proposed in recent years [14], [15], [16]. Reducing the number of layers and performing a matrix factorization to design a lightweight model are effective means to improve model efficiency [11]. Due to the limitations of the parameters and the manner of training from scratch with raw data, it is difficult to obtain a high-performance counting model. The network architecture search optimizes the block configuration to realize the model compression, while it is time-consumed to design a common searching strategy due to the diverse networks and the high-dimensional search space. Knowledge distillation (KD) is an effective method to compress the model, which transfers the knowledge from the heavy teacher model to the lightweight student model to make the small model perform better than the large model. What knowledge is extracted from the teacher and how to transfer the knowledge from the teacher to the student are critical issues in KD. Most KD methods are designed for image classification, which cannot be transferred directly to pixel-level tasks such as crowd counting. To the best of our knowledge, SKT is the first attempt to use KD to obtain lightweight counting models [17]. It considers the intra-layer pattern and inter-layer relation, uses cosine similarity and L1 loss to transfer the information embedded in the feature map from the teacher model to the student model, and achieves good performance on the VGG-based model. This paper aims to further explore the method of the lightweight counting model based on KD to improve the generalization ability of the student model.

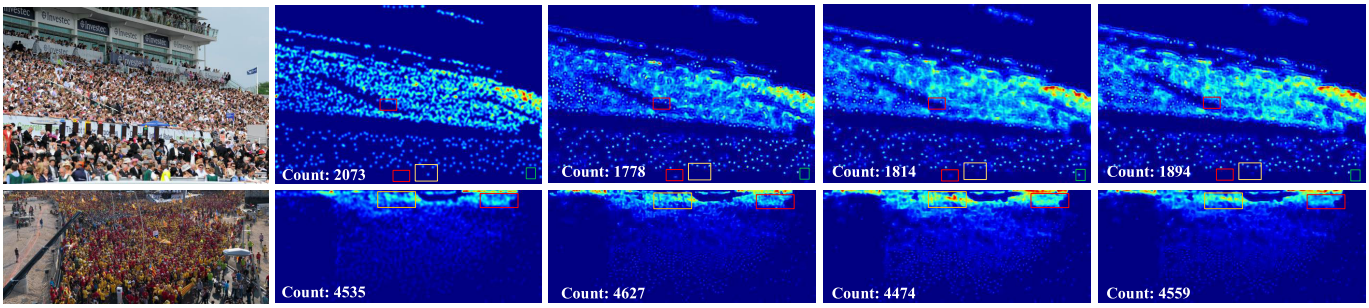

Fig. 1. Visualization of density map. (a) Original image. (b) Ground truth (GT). (c) The results calculated by the teacher model. (d) The results calculated by the student model after KD from the teacher model. (e) The results calculated by the student model after SKD.

During executing KD, there are strict limitations for the knowledge to be transferred in the teacher model. It is better to only transfer the more credible knowledge to boost the positive update of the student model. As shown in Fig. 1, we visualize the density maps generated by the teacher and student models based on the classic counting model known as Bayesian Loss (BL) [18]. In the first row, compared to the ground truth (GT) in Fig. 1 (b), the teacher model learns the wrong information marked with the red and green boxes as shown in Fig. 1 (c), and the people who exist or non-exist in GT are wrongly perceived by the teacher model. By performing a knowledge-transfer strategy from the teacher model to the student model, as proposed in this paper, shown in Fig. 1 (d), the student model follows the wrong direction of the teacher model. Although the GT is also utilized during KD, the error estimation and bias of the teacher model on some samples lead to the wrong guidance to the student model. Therefore, it is necessary to transfer the knowledge from the teacher to the student at an early stage of the training and make the student model explore more external information based on GT without disturbance by the teacher model in the later stage.

To address the above issues, we propose a dual-knowledge distillation (DKD) framework to train a lightweight crowd-counting model based on KD. In the first stage, we perform the knowledge transfer from the teacher to the student to obtain the initialized student model based on the soft information and hard labels. In the second stage, to ensure the accurate information learned by the student model, a self-knowledge distillation (SKD) mechanism is proposed. It allows the student to fine-tune the model independently, based on the GT and its own information generated by the GT, removing the limitation of strong supervision by the teacher and learning correct knowledge under the supervision of the GT. In Fig. 1 (e), after completing the self-distillation stage, the wrong information originally learned from the teacher model is corrected in the DKD model. In the second row, the regions in the yellow box are also corrected by the DKD model that are not correctly annotated in GT due to the congested effects.

In our work, inspired by [19], the pixel-wise tasks with a large backbone, are divided by the encoder and classifier to represent the distribution and observations. According to the structure of the counting network, we deem the backbone as the feature extractor and the last layer of backbone as the projector and the counting regressor is used to output results respectively. In the first stage of KD from the teacher model to the student model, the approximate GcNet [20] is used to extract the global and adaptive features of the projector. The optimal transport distance (OTD) is used to match the features, including intermediate feature maps and the last feature map of the projector. In the second stage of SKD, the intermediate feature maps match with the context information of the projector extracted by the approximate GcNet. In addition, the correlation loss is calculated between the projector and the estimated final density map. Through this hierarchical self-distillation mechanism, the student model adaptively learns the features from different views to achieve the self-correction based on the soft information and hard labels. After two stages, the student model completes the training utilizing the prior knowledge learned by the teacher model and fine-tuning it to correct the previous errors. In summary, our contributions are:

1) We propose a general DKD framework for crowd counting to generate lightweight and efficient counting models. There are two stages in DKD, including the knowledge transfer from the teacher and self-distillation to correct and fine-tune the student model.
2) We develop the idea of adaptive perspective distillation into the counting task to learn target map through hierarchical distillation, which accelerates the learning ability of the intermediate feature maps.
3) In the process of knowledge transfer, we formulate the OTD to measure the feature representation, improving the matching degree of the feature distribution.
4) To demonstrate the superiority of the proposed method, extensive experiments on four challenging benchmarks show that DKD can effectively train high-performance and lightweight student models.

The rest of this paper is organized as follows. Section II reviews the related work on crowd counting and KD. Section III introduces the proposed DKD framework with two stages. Section IV presents the experiments and the results on several benchmarks. And we conclude our work in Section V.

II. Related Work

In this section, we review previous work from two aspects related to our study, i.e., crowd counting and KD.

A. Crowd Counting

Researchers regard the crowd counting task as counting regression, obtaining the crowd estimation based on hand-craft

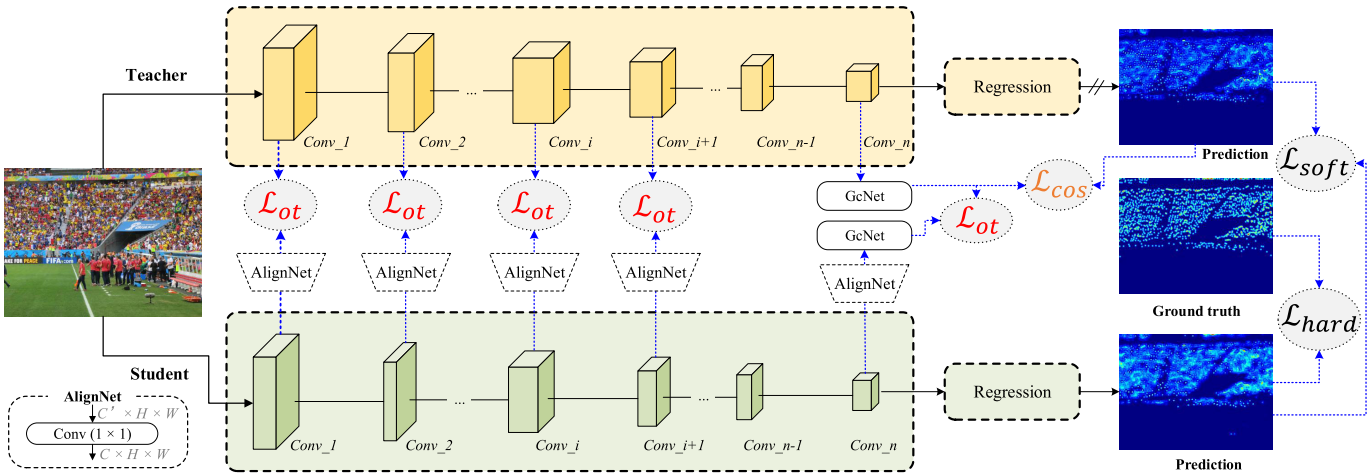

Fig. 2. Overview of the knowledge transfer from the teacher model to the student model. There are four loss functions. (a) The $\mathcal{L}_{ot}$ measures the distance between feature maps of the teacher and student. (b) The $\mathcal{L}_{cos}$ measures the distance between the projector and the target of the teacher model. (c) The $\mathcal{L}_{soft}$ measures the distance between the outputs of the student and teacher. (d) The $\mathcal{L}_{hard}$ measures the outputs of the student and GT.

features [21], [22] and automatic deep networks [12], [23], [24], [25] in the previous studies. These methods aim to obtain the number of people but ignore the head location in the scene. The detection-based method is developed to detect the face location to obtain the crowd estimation [26], [27]. The density-map regression method processes the head annotation with the adaptive two-dimensional Gaussian distribution [28], which reduces the prediction difficulty and preserves the spatial attribute. The multi-scale and multi-level feature representation methods based on density map regression are explored to handle the issue of perspective distortion [29], [30]. Since the Mean Squared Error (MSE) loss function of these methods weaken the precise location of the overlapping heads, some methods based on the point-regression focus on designing the appropriate loss function [31], [32], [33], [34]. The BL model designs a density-contribution probability model for each point as the supervision information [18]. The DMC proposes the OTD to measure the similarity between the normalized density map, thereby improving the generalization error bounds [31]. The P2PNet proposes density-normalized average precision, directly predicting a set of point proposals and matching these proposals based on the Hungarian algorithm [32]. Meanwhile, the demand for lightweight counting models is increasing [35], [36], [37], [38]. The TED-Net aggregates features in different encoding stages via multiple decoding paths, achieving a performance improvement with fewer parameters [39]. The MobileCount is a computation-efficient encoder-decoder architecture based on the MobileNetV2 to boost the performance with a small increase of FLOPs [40]. The Lw-Count is also an effective encoding-decoding lightweight counting network to make an optimal trade-off between counting performance and running speed [41]. The SKT is a general-knowledge transfer framework that conducts effective distillation, achieving at least a 6.5× speedup on a GPU and a 9.0× speedup on a CPU, while maintaining a competitive counting performance [17]. The ECCNAS proposes an efficient network-architecture search framework for crowd counting [42]. In this paper our approach based on KD reduces the space and time complexity of the existing counting models while ensuring the least loss of the counting performance.

B. Knowledge Distillation

KD is one of the effective means to obtain small models from large models. It was proposed by Hinton [43], which makes a teacher model transfer soft labels to the student model in the classification tasks. The FitNets distills the knowledge of intermediate layers to guide the student model in the semantic segmentation [44]. The AT forces the student model to imitate the attention map of the teacher network [45]. The AB proposes an activation transfer loss to make the student network learn the separation boundary between the activation and deactivation regions formed by each neuron in the teacher network [46]. The FSP transfers the knowledge by computing the inner product between the features of the two layers [47]. The PKT uses different kernels to estimate the probabilities of the teacher model and student model as a divergence measurement in knowledge transfer [48]. In pixel-level tasks, the KA captures long-range dependencies in semantic segmentation tasks by computing non-local interactions across the image [49]. The DeFeat decouples the detector, divides it into neck and classification features, and obtains an efficient student detector by assigning weights to the feature maps of different regions [50]. However, these reports ignore the effect that the misinformation generated by the teacher model leads to the wrong guidance for the student model, which has promoted the research of the SKD [51], [52]. In [51], the student model extracts knowledge within the network, enabling the shallow layers to learn the knowledge of the deep layers. The PS-KD proposes progressive SKD, which achieves better accuracy for image classification, object detection, and machine translation by gradually extracting the soft and hard knowledge from the target [53]. In this paper, we combine the advantages of teacher-to-student knowledge transfer and the self-distillation of the student to obtain an efficient model.

III. Method

The proposed DKD includes two stages, one of which is knowledge transfer from the teacher to the student as shown in Fig. 2, and the second stage is SKD based on the student and GT as shown in Fig. 5. In this section, we elaborate on the details of DKD.

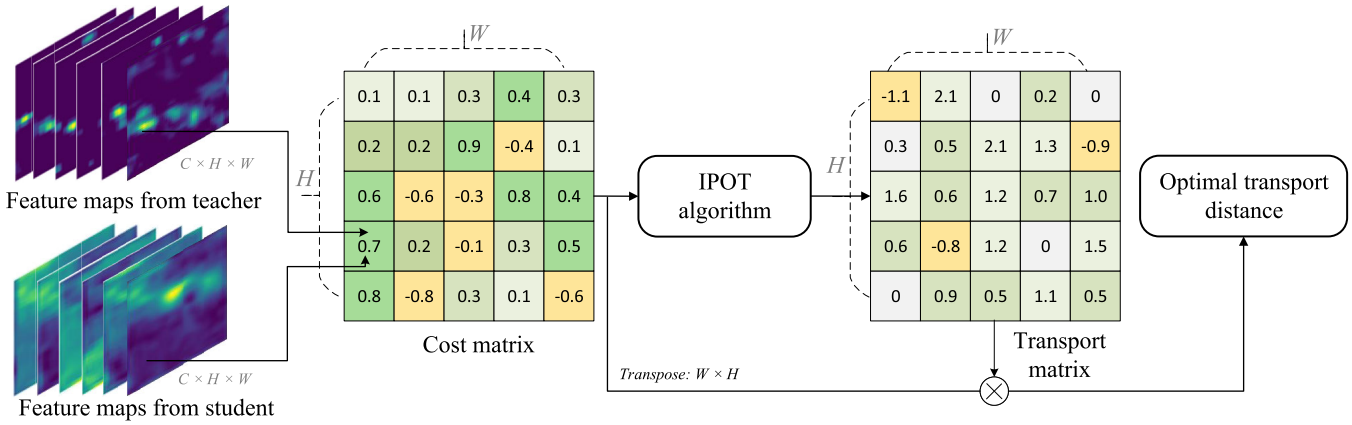

Fig. 3. Illustration of OTD to measure the distribution distance between features maps of teacher and student. The cost matrix is obtained and the IPOT algorithm is used to obtain the transport matrix, and then the optimal transport distance is calculated.

A. Knowledge Transfer From Teacher to Student

The teacher model and the student model have the approximate network structure, and the student model is compressed by reducing the number of channels in each layer of the large model according to the compression ratio. Following [17], we use the Channel Preservation Rate (CPR) as the lightweight indicator. For example, when implementing 1/4 CPR on the large model, if there are C channels in a layer of the large model, the number of channels in the student model is $C/4$. We fix the CPR, indicating that each layer has the same proportion of pruned channels.

Fig. 2 shows an overview of knowledge transfer from the teacher to the student. The layers of KD include intermediate feature maps, projector, and final map. The teacher model is a pre-trained network, whose parameters are frozen during the training of the student. The student is a smaller pruned network with a fixed CPR. AlignNet, consisting of a 1×1 convolutional layer, is used to align the number of channels of the student with the teacher model. The interpolation is used to unify the size of the feature map generated by the teacher and the student. Taking the BL model as an example, the network structure used in the BL model is VGG19, divided into a feature extractor and a regressor module. There are four loss functions to perform the KD for the student model.

For the intermediate feature maps of the feature extractor, the OTD is used to measure the discrete distribution gap, as shown in Fig. 3. A solution to minimize the global transmission cost can be solved and the dynamic transfer between the student and teacher is realized. The sizes of the feature map generated by the teacher and student are $C \times H \times W$ and $C' \times H \times W$. The feature maps of the student are aligned by the AlignNet to keep consistency with the teacher. We consider the transmission cost of the two feature maps over the spatial location assuming that the crowd numbers are distributed in a closed form in the spatial region, and thus the transport distance is calculated at the spatial level. Specifically, there are two probability distributions $f_s \sim \mu$ and $f_t \sim \upsilon$, representing the discrete distribution of interest in the feature map. It can be interpreted as an efficient way to transfer the probability knowledge from υ to μ, which is fitted for every training sample. Without considering the approximation loss, we assume $\sum_i \mu_i = 1$ and $\sum_j \upsilon_j = 1$. The transport matrix is $\pi \in \mathbb{R}^{w\times h}$ whose row and column marginals match μ and υ, and they obey $\sum_i \pi_{ij} = \mu_i$ and $\sum_j \pi_{ij} = \upsilon_i$. In this study, it is interpreted as making the elements transfer from the location $i \in w$ to the location $j \in h$ according to the transport strategy. During training, we make sure $W = H$ so that the width and height of the feature map are equal. Since this transport plan is not unique, it is necessary to employ a better way to solve it. To find the minimum transfer distance under a given cost function $\mathcal{L}_{ot}(\theta_s)$, that is

$$\mathcal{L}_{ot}(\theta_s) = \sum_{ij} \pi^*_{ij} \cdot C(f_s, f_t) = \inf_{\pi\in(\mu,\upsilon)} \sum_{ij} \pi_{ij} \cdot C(f_s, f_t), \tag{1}$$

where $C(\cdot)$ is a cost function making the knowledge transfer between the two latent feature vectors. We use the cosine similarity to calculate the structure similarity between the student and teacher models in the spatial dimension as follows:

$$C(f_s, f_t) = 1 - \frac{f_s^\top \cdot f_t}{\| f_s \|_2 \| f_t \|_2}, \tag{2}$$

where $\| \cdot \|_2$ is the Euclidean norm. With reference to [54], we use the inexact proximal point method for the optimal transport (IPOT) algorithm to solve the optimal transport matrix π^*_{ij} by Sinkhorn-based proximal point iterations. Compared with the Sinkhorn algorithm, this method speeds up the learning process and makes the training more stable. The details of the IPOT are shown in Algorithm 1. Lines 3-9 iteratively solve the transport plan without back-propagating the gradient, the time complexity of the method based on the Envelope Theorem [55] is low. After the result reaches convergence, Line 10 multiplies the allocation scheme by the cost matrix and computes the trace of the matrix to obtain the transport distance.

The projector as the last feature map of the feature extractor is similar to the target density map. The OTD captures fine-grained information distribution at the pixel level of the feature map. However, this approach compromises the global context information. We propose to extract global relations using GcNet [20], as shown in Fig. 4, and perform KD from the teacher model to the student model. This is a lightweight module that can be plugged into the intermediate layers of

Algorithm 1 IPOT Algorithm

Input: Feature maps $\{f_s^i\}_{i=1}^n$ and $\{f_t^j\}_{j=1}^n$, probability vectors μ and υ, β, and the size of circle step t and k.

Output: Optimal transport distance l.

$\sigma = \frac{1}{n}$, $\pi^1 = 11^\top$

$C_{ij} = c(f_s, f_t)$, $A_{ij} = \exp^{-\frac{C_{ij}}{\epsilon}}$.

for t = *1,2,⋯* **do**

  $Q = A \odot \pi^t$ // $\odot$ is the Hadamard product.

  for k=*1,2,⋯* **do**

    $\delta = \frac{\mu}{nQ\sigma}$, $\sigma = \frac{\mu}{nQ^\top \delta}$

  end

  $\pi^{t+1} = \mathrm{diag}(\delta) Q \mathrm{diag}(\sigma)$

end

$l = \mathrm{Tr}(\pi^\top \otimes C)$ // $\otimes$ is the Frobenius dot-product.

Return l

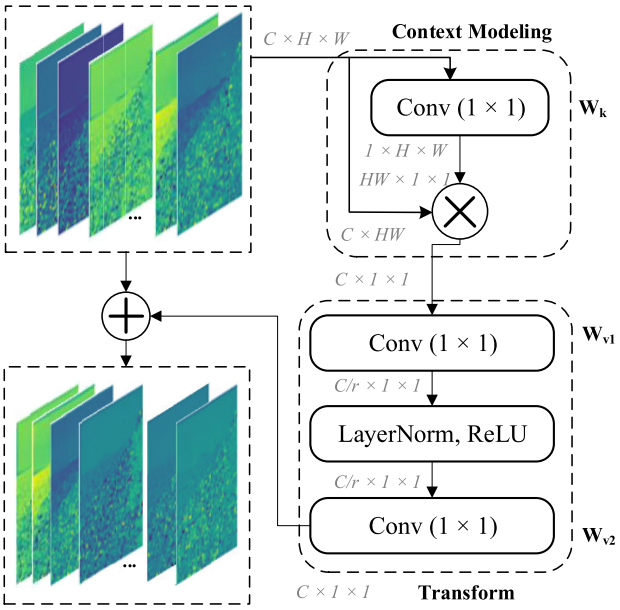

Fig. 4. GcNet, used to extract global information from feature maps [20]. The inputs are the feature maps of the student and teacher models.

deep-learning models to capture long-range dependencies of the salient regions with a low computational cost. Both of the pieces of global context information of the feature maps are extracted based on GcNet, after which the OTD loss between teacher and student is calculated. For the feature map f, it is processed by GcNet, expressed as

$$G(f) = f + W_{v2}\mathrm{ReLU}(\mathrm{LN}(W_{v1}\sum_{j=1}^{N_p}\frac{\exp^{W_k x_j}}{\sum_p^N \exp^{W_k x_m}} x_j)), \quad (3)$$

$$G(f) = f + W_{v2}\mathrm{ReLU}(\mathrm{LN}(W_{v1} \cdot W_k \cdot x)), \quad (4)$$

where W_{v1}, W_{v2} and W_k are CNNs, LN is the layer normalization, and ReLU is the activation function. Unlike GcNet [20], we do not perform Softmax processing features in the context-modeling module. In particular, both AlignNet and GcNet are inserted as attachments in the process of calculating the loss function. These are not part of the counting network and are removed during inference.

To extract the global context information of both the student model and the teacher model, there are two GcNet blocks plugged into the teacher and student. Due to the parameters of the teacher model being frozen during the knowledge transfer, an extra loss function is formulated to optimize the parameters of the GcNet plugged into the teacher model. The optimization goal of the counting model is to obtain high-quality density maps and accurate counts, so we force the global information of the teacher network to maintain structural similarity with its density map predictions. In this way it cannot only be used to update the parameters of GcNet, but also guide the feature extractor to generate high-quality density maps in the early stage. The loss function between the target density map of the teacher and the context information of GcNet is

$$\mathcal{L}_{tea}(\theta^t_{GcNet}) = 1 - \frac{G(f_t^\top) \cdot D_t}{\| G(f_t) \|_2 \| D_t \|_2}, \quad (5)$$

where D_t is the density map predicted by teacher model. Since in the regressor module, the convolutional operation cannot reduce the size of feature map, the size of $G(f)$ is the same as the target density map, which is calculated with cosine loss expressed as $\mathcal{L}_{cos}$. For GcNet embedded in the student model, its parameters are learned in the back-propagation of the student model with all losses.

During the training of the counting model in general, the model uses GT as the supervised information. In the KD task, since the teacher model has been trained on the dataset, the distribution information of the dataset and the characteristics of the task have been stored in their own parameters. Therefore, the prediction results of the teacher model are used as the supervision information of the student model. Meanwhile, there are manual labeling errors or bias in insufficient GT for some samples. For example, in the second image in Fig. 1, the part occluded by the wire in the GT is accurately predicted by the teacher model. This inspires us to utilize the results estimated by the teacher model to assist in guiding the learning of the student model. The predicted density maps can be treated as knowledge called the soft GT, defined as

$$\mathcal{L}_{soft} = \| D_s - D_t \|^2, \quad (6)$$

where D_s and D_t are the density maps estimated by the student model and teacher model. Meanwhile, the estimated results of the student and GT are defined as the hard GT, and the supervision loss function is expressed as

$$\mathcal{L}_{hard} = \varphi(D_s, D), \quad (7)$$

where D is the GT and φ is the loss function defined in the original counting model, which is determined by the selected model. Finally, we use the following total loss to optimize the model for knowledge transfer from the teacher to the student:

$$\mathcal{L}_{total1} = \lambda_1 \sum_{i=1}^{m} \mathcal{L}^i_{inter\_ot} + \lambda_2 \mathcal{L}_{proj\_ot} + \lambda_3 \mathcal{L}_{cos} + \lambda_4 \mathcal{L}_{soft} + \lambda_5 \mathcal{L}_{hard}, \quad (8)$$

where $\mathcal{L}^i_{inter\_ot}$ and $\mathcal{L}_{proj\_ot}$ represent the OTD between the intermediate feature map and projector, λ_1, λ_2, λ_3, λ_4 and

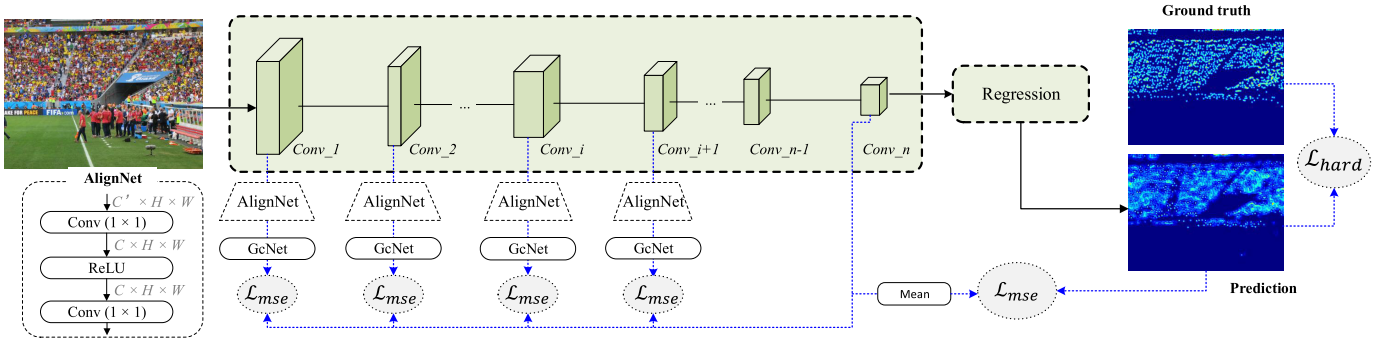

Fig. 5. Overview of SKD. There are three loss functions. (a) The $\mathcal{L}_{inter\_mse}$ measures the distance between the intermediate feature map and projector. (b) The $\mathcal{L}_{proj\_mse}$ measures the distance between projector and final map. (c) The $\mathcal{L}_{hard}$ measures the distance between final results and GT.

λ_5 are the balance weights of the losses and m is the number of intermediate feature maps.

B. Self-Knowledge Distillation via Student

To overcome the strong dependence of the students on the teacher and their distribution gaps in knowledge transfer, we propose a self-distillation method, by which the student model uses the information generated by itself and the GT to boost the update. Some studies force the intermediate feature maps to learn the final results to keep the consistency of the global information in the field of image classification [56]. Inspired by these studies, in the process of SKD on crowd counting, we use the output of the student model as supervision information to guide the learning of the intermediate feature maps and accelerate the model convergence. Since the regressor module in the counting network gradually generates the feature map to approximate the target map, we choose the projector to realize the target-awared map generation of the intermediate layer. Meanwhile, we minimize the difference between the predicted density map and the GT, and transfer knowledge from the predicted final map to the projector. Through the hierarchical distillation method, the knowledge in the deeper layers of the network is distilled into the shallow layers, avoiding the early bad guidance introduced by the teacher model.

Fig. 5 shows the overview of the SKD via the student. To keep consistency of the dimensions, we use AlignNet to match the channel number of the intermediate layers with the projector. AlignNet consists of a network module of Conv (1 × 1)-ReLU-Conv (1 × 1). Since the projector is a high-dimensional feature map generated by the deeper layer, it contains abundant global and local information about images. The shallower layers generate the texture features and the generalization features. Intuitively, it is difficult for them to keep strict consistency. Benefiting from the advantage of GcNet, which aggregates the query-independent global context for capturing long-range dependency, we apply the GcNet block on every intermediate feature map to extract the global information. The soft method makes the shallower feature maps processed by GcNet match with the deep feature to avoid the over-learning of the student model on the shallower layers. The distance between the projector and the feature maps of the intermediate layers is measured by MSE, defined as

$$\mathcal{L}^{i}_{inter\_mse} = \| G(f^{i}_{s}) - f^{*}_{s} \|^2, \tag{9}$$

where f^{*}_{s} is the projector, f^{i}_{s} is the i_{th} feature map of the intermediate layer. In addition, by averaging f^{*}_{s} in the channel dimension, it is consistent with the dimension of the target map, and the loss function of the MSE is defined as

$$\mathcal{L}_{proj\_mse} = \| D_s - \phi(f^{*}_{s}) \|^2, \tag{10}$$

where D_s is the density map estimated by student model and $\phi(f^{*}_{s})$ is average operation for feature map f^{*}_{s}. The hard loss is calculated with the estimated results of the student and the GT. Finally, the total loss function is used to optimize the student model, defined as

$$\mathcal{L}_{total2} = \alpha_1 \sum_{i=1}^{m} \mathcal{L}^{i}_{inter\_mse} + \alpha_2 \mathcal{L}_{proj\_mse} + \alpha_3 \mathcal{L}_{hard}, \tag{11}$$

where α_1, α_2, and α_3 are the balance weights of the different losses and m is the number of intermediate feature maps.

C. Dual Knowledge Distillation

Knowledge transfer from the teacher to the student aims to match the distribution from μ to υ based on OTD, achieving initial distillation for the student model. Meanwhile, SKD makes the multi-level knowledge squeeze from the student model. Section III.A optimizes the distribution matching and Section III.B minimizes inner target estimation to obtain a more accurate density map.

Knowledge transfer and self-distillation are designed into two stages, and self-distillation is complementary to the first stage. The $\mathcal{L}_{total1}$ aims to minimize local and global representations to reduce the gap between the two feature spaces. And $\mathcal{L}_{total2}$ is designed to make the compliment and correction based on the first stage. Finally, an efficient student model with a smaller size and better performance is obtained.

IV. Experiments

To demonstrate the effectiveness of the proposed framework, we evaluate the DKD on the four public datasets based on the typical network architectures in this section. We introduce experimental settings, compare our method with the state of the art, and in the last perform some ablation studies.

A. Experiment Settings

1) Datasets: The four datasets are used to evaluate our DKD framework, including the ShanghaiTech Part_A (SHHA) [57] and Part_B (SHHB) [57], UCF_CC_50 [58], and UCF-QNRF [59] with various resolutions and crowd densities. Meanwhile, to further demonstrate the robustness of our proposed DKD model, FSC-147 object counting dataset [60] which has 147 object categories such as vehicles, animals and fruits is used to evaluate the DKD framework.

2) Evaluation Metrics: To evaluate the counting performance, we use the Mean Absolute Error (MAE) and MSE as the metrics, which are defined as:

$$MAE = \frac{1}{N}\sum_{i=1}^{N} \| D_s^i - D^i \|, \tag{12}$$

$$MSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \| D_s^i - D^i \|^2}, \tag{13}$$

where N is the number of images in the testing dataset, D and D_s are the GT and estimated density map.

To evaluate the model size and the amount of computation, the number of parameters in the convolutional network is:

$$\text{Params} = (K_h \cdot K_w \cdot C_{in} + 1) \cdot C_{out}, \tag{14}$$

where C_{in} is the number of input channels, K_h and K_w are the convolution kernel size, $+1$ represents the amount of operation of the bias term, and C_{out} is the number of output channels. The computational number of convolutional layers is measured with floating-point operations (FLOPs):

$$\text{FLOPs} = 2HW(C_{in} \cdot K_h \cdot K_w + 1) \cdot C_{out}, \tag{15}$$

where H, W and C_{out} are height, width, and the number of channels for the feature map, K_h and K_w are CNN kernel size, and C_{in} is the number of channels for input, $+1$ represents the amount of operation of the bias. In this paper, the units of Params and FLOPs are million (M) and Giga (G). We also evaluate the efficiency of models on the CPU and GPU to obtain the inference time and frames per second (FPS). And the units of time on the CPU and GPU are seconds and milliseconds.

3) Network Architecture: To evaluate the effectiveness of the proposed framework, we use a classical counting model called the BL model, which achieves better performance on some challenging benchmarks. The model employs a typical image-classification network VGG19 as the backbone [64], and the last connected layers are removed and replaced with a regression module with three 3×3 convolutional layers of 256 and 128 output channels, and 1×1 convolutional layer to regress the feature map to the density map. The student model is compressed according to the CPR performed on every layer, except for the last layer of the regressor. In addition, we apply our DKD on other models such as CAN [61], SFCN [62], and DMC [31] and their compressed student models to validate generalization and robustness of proposed method.

4) Implementation Details: Before making the KD for the student model, the teacher model is trained on the four datasets. For the BL model, the backbone of the teacher model is initialized with the results trained on ImageNet, and the regressor is initialized with the Kaiming initialization. The Bayesian loss is used to optimize the parameters of the model proposed in [18]. The images are cropped as the fixed size for training and the crop size is 256×256 for SHHA due to the smaller resolution and 512×512 for SHHB, UCF_CC_50, and UCF-QNRF. The other parameters involved in the loss function keep consistency with the BL model. Other teacher models in the generalization experiments are trained based on the parameters of the published versions.

For the configurations of the student model, according to the experimental setting, we set the different CPRs to 1/2, 1/3, 1/4 and 1/5 for the VGG19. In the process of KD for the student model, the Adam optimizers with an initial learning rate of $1e-4$ and $1e-5$ are used to update the student model. For the attached network, such as AlignNet and GcNet, for the first stage of knowledge transfer from the teacher to the student, the initial learning rate of the Adam optimizer is $1e-3$ and for the second stage of SKD, the SGD optimizer is utilized with the initial learning rate $2e-2$ and the momentum 0.98. The weight decay is $1e-4$ in the two stages. To compute the OTD, the β, k and t for the intermediate feature maps are 0.5, 3 and 3, and 0.6, 6 and 3 for the last feature map of the feature extractor. In particular, the feature maps after the Maxpooling layer are used as the intermediate feature maps in our study. The λ_1, λ_2, λ_3, λ_4 and λ_5 are 100.0, 100.0, 1.0, 1.0 and 1.0, while the α_1, α_2 and α_3 are 1.0, 1.0 and 1.0.

B. Comparison With Crowd-Counting Methods

1) Performance Comparison: We compare the performance of our DKD framework with the recent counting models on four datasets, as shown in Table I. The baseline means to train the student model based on the GT, and the performance is heavily degraded compared with the original BL model. After going through the first stage of the KD from the teacher to the student, the student models perform well on the four datasets. Furthermore, the DKD improves the performance of student model by learning knowledge by itself.

There is still a large gap between the performance of our model and large models, such as D2C [4], P2PNet [32], and MAN [2]. However, in the same training environment, the training time of D2C, MAN and P2P is approximately 72 hours, 63 hours and 55 hours, while the proposed DKD model only need 20 hours to complete two training stages. It can be found that DKD model also have better training efficiency among these larger models. The TransCrowd is a transformer-based count-regression method that first applies the transformer to the crowd counting, but with a large number of parameters. Compared with the lightweight models, our model achieves comparable performance on the SHHA dataset, but is weaker than SKT and ECCNAS, the architectures of which are all compressed based on the BL model. The complete BL model with a size of 21.50M has an MSE of 103.2 and our model with 1.35M achieves 103.0 stronger than the original model. As for the SHHB dataset, our model surpasses all the existing methods achieving state-of-the-art results for the lightweight models with an MAE of 7.4 and an MSE of 12.7. The results are very close to those of the original model, but the model size is only 6.28% of the original model. The tiny UCF_CC_50 dataset achieves the best

TABLE I

PERFORMANCE COMPARISON ON FOUR DATASETS. THE BASELINE REPRESENTS THE RAW STUDENT MODEL TRAINED ONLY WITH GROUND TRUTH. KD IS THE MODEL OF THE FIRST STAGE OF DKD FROM THE TEACHER TO THE STUDENT, WHILE DKD IS THE MODEL OF THE SECOND STAGE OF SKD

Methods	Year	Params	SHHA		SHHB		UCF_CC_50		UCF-QNRF	
			MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
CSRNet [9]	CVPR18	16.26	68.2	115.0	10.6	16.0	266.1	397.5	145.5	233.3
CAN [61]	CVPR19	18.89	62.3	100.0	7.8	12.2	212.2	243.7	107.0	183.0
BL [18]	ICCV19	21.50	61.5	103.2	7.5	12.6	229.3	308.2	87.7	158.1
SFCN [62]	CVPR19	47.83	64.8	107.5	7.6	13.0	214.2	318.2	102.0	171.4
DMC [31]	NeurIPS20	21.50	59.7	95.7	7.4	11.8	211.0	291.5	85.6	148.3
LSC-CNN [63]	T-PAMI20	35.08	66.4	117.0	8.1	12.7	225.6	302.7	120.5	218.2
D2C [4]	TIP21	35.77	57.2	93.0	6.3	10.7	182.1	254.9	81.7	137.9
P2PNet [32]	ICCV21	18.34	52.7	85.1	6.3	9.9	172.7	256.1	85.2	154.5
TransCrowd [12]	SCIC22	89.10	66.1	105.1	9.3	16.1	272.2	395.3	97.2	168.5
MAN [2]	CVPR22	30.90	56.8	90.3	-	-	-	-	77.3	131.5
MCNN [57]	CVPR16	0.13	110.2	173.2	26.4	41.3	377.6	509.1	277.0	426.0
SANet [11]	ECCV18	0.91	67.0	104.5	8.4	13.6	254.8	334.9	152.6	247.0
TEDNet [39]	CVPR19	1.63	64.2	109.1	8.2	12.8	249.4	354.5	113.0	188.0
LCNet [37]	ICIP19	0.86	93.3	149.0	15.3	25.2	326.7	430.6	-	-
PCC-Net [14]	TCSVT19	0.55	73.5	124.0	11.0	19.0	240.0	315.0	148.7	247.3
MobileCount [40]	Neuro20	3.40	84.4	135.1	8.6	13.8	284.8	392.8	127.7	216.5
CCNN [38]	ICASSP20	0.07	88.1	141.7	14.9	22.1	-	-	-	-
SKT [17]	MM20	1.35	62.7	102.3	8.0	13.1	235.9	322.3	96.2	156.8
ECCNAS [42]	TOMM22	3.88	62.0	110.9	7.5	12.9	223.1	293.8	91.2	158.9
Baseline	-	1.35	88.4	145.5	12.3	19.8	241.1	331.9	135.6	224.7
Our KD	-	1.35	67.3	106.0	8.5	13.4	226.5	318.4	94.0	155.8
Our DKD	-	**1.35**	**64.4**	**103.0**	**7.4**	**12.7**	**210.3**	**283.8**	**91.7**	**150.1**

results, which are much higher than the other models. The MAE and MSE are better than about 10, compared with the ECCNAS model. As the prior knowledge, the BL model is good at counting on the dense scenario, which performs better on UCF-QNRF dataset. According to the evaluation on UCF-QNRF, the MAE of our method is a little weaker than the ECCNAS, the results of which are close to the original model. In addition, the results of the MSE are better than the original model. In the following sections, some experiments demonstrate our method performs better than the ECCNAS model when the model size is increased, but still smaller than 3.88M. Interestingly, our DKD performs better than the teacher model on the UCF_CC_50 and UCF-QNRF datasets. During the KD, the teacher model provides more soft information to accelerate the training of the student model based on IPOT and GcNet. In the second stage of DKD, the SKD offers more pseudo-annotation except for the GT, which avoids the disturbance caused by noisy labels and teacher model. With the proposed two stages, our DKD achieves comparable performance even better than the teacher model with fewer Params and FLOPs. In addition, as can be seen, our proposed DKD performs well on the SHHB, UCF_CC_50, and UCF-QNRF, but is a little weak on the SHHA dataset. The average resolution of SHHA is 589 × 868, and its crowd density is 9.81, showing there are many people in the small-scale images. Meanwhile, the crowd densities of other datasets such as SHHB, UCF_CC_50 and UCF-QNRF are 1.57, 2.11 and 1.39 with average resolutions of 768×1024, 2101×2888 and 2013×2902 [65]. Our DKD method is weaker than SKT and ECCNAS for processing small-resolution images, but it is suitable for counting tasks with high-resolution cameras that have strict requirements on latency and model size. It shows that our method is not enough to transfer fine-grained texture information in knowledge extraction and feature representation, but is good at learning and understanding global information of various images. The student model distilled by DKD, which uses a small number of parameters, but outperforms the overall performance of the teacher model, is popular in smart cities where high-definition cameras are deployed.

2) Inference Efficiency Comparison: To verify the superiority of DKD in terms of inference efficiency, we compare our method with the existing counting models on the UCF-QNRF dataset. We measure the number of parameters, FLOPs, the inference time on a CPU and GPU, and the counting speed of FPS. The results are shown in Table II. The average resolution of images in the UCF-QNRF dataset is 2013×2902. The models such as D2C, MAN, BL, CAN and P2P have larger model sizes and computations, especially the Params and FLOPs of D2C model are 35.77M and 5303.0G. And the counting accuracy of P2P model is better than other models, but the inference time of P2P on CPU is obviously slower than some lightweight models. It is observed that the MCNN is the most lightweight model with a number of parameters of 0.13M. The running time of MCNN on the CPU and

TABLE II

THE INFERENCE EFFICIENCY OF COUNTING MODELS ON UCF-QNRF DATASET. THERE ARE SIX GOLD 6252 CPUS AND A TESLA V100 GPU USED TO MAKE INFERENCES

Methods	Params	FLOPs	CPU		GPU	
			Time	FPS	Time	FPS
D2C [4]	35.77	5303.0	48.7	0.02	4.5	222.2
MAN [2]	30.90	2176.0	89.9	0.01	3.5	285.7
BL [18]	21.50	1759.4	51.9	0.02	3.1	322.6
CAN [61]	18.89	2601.9	68.0	0.01	6.2	161.3
P2P [32]	18.34	2196.7	40.5	0.02	3.3	303.0
CSRNet [9]	16.26	1226.8	37.8	0.03	3.0	333.3
MobileCount [40]	3.40	33.2	37.9	0.03	10.5	95.2
SANet [11]	1.39	375.0	55.7	0.02	8.2	122.0
SKT-BL [17]	1.35	155.3	15.2	0.07	3.0	333.3
MCNN [57]	0.13	115.1	14.2	0.07	2.7	370.3
Our DKD	**1.35**	**111.6**	**14.4**	**0.07**	**2.9**	**344.8**

TABLE III

THE GENERALIZATION OF THE DKD, WHICH IS PERFORMED ON THE DIFFERENT TEACHER AND STUDENT MODELS ON THE UCF-QNRF DATASET. BAYESIAN*-1/4 AND DMC ARE FROM VGG19 STRUCTURE BUT WITH DIFFERENT ALIGNMENT LAYERS AND SUPERVISION LOSSES

Teacher		Student		MAE	MSE
Model	Params	Model	Params		
VGG19 [18]	21.50	AlexNet	2.50	123.3	202.7
		SFCN-1/4	29.70	103.0	179.5
		Bayesian*-1/4	1.35	101.2	160.2
		DMC-1/4	1.35	93.3	152.9
CAN [61]	18.89	CAN-1/4	1.18	118.7	200.3

GPU are both the shortest among these models. It is noted that the SKT-BL and our DKD have the same number of parameters, but the speed of the DKD is a little faster than SKT-BL. Interestingly, the number of parameters of SANet, SKT-BL and DKD are approximate, but the FLOPs of SANet are larger than the two models, which takes more inference time. Our DKD model achieves comparable performance with an inference time of 14.4 seconds and 2.9 milliseconds on the CPU and GPU. The FPS on the GPU is 344.8, which meets the real-time processing requirements of high-definition images. By comprehensively considering the counting accuracy and running speed, our DKD can realize efficient counting in a real-time system.

3) The Generalization of the DKD: To verify the generalization of the proposed DKD, we apply it on the different structures of teacher and student based on different GT supervisions. The results are shown in Table III. For Alexnet and SFCN transferring knowledge from VGG19 based on MSE loss, there is a small performance decrease compared with the original model. Furthermore, we adapt the 1/4 VGG19 as a student model but only align the part of the layers that have different channel numbers and feature sizes with the teacher model based on Bayesian loss [18]. It can be seen that the performance with the MAE and MSE of 101.2 and 160.2 is weaker than the way of completely aligning the feature map with the MAE and MSE of 91.7 and 150.1. Distributed matching loss [31] is also used as the supervision method, and the compressed DMC model shows better results with the MAE and MSE of 93.3 and 152.9, which is approximate with the original teacher model. In addition, we perform DKD on other teacher models with different structures. The CAN model is a multi-scale counting model with multiple receptive field sizes to adaptively encode the contextual information of the image. By compressing the parameters of the CAN as 1/4 with the original model to be as the student model, the MAE and MSE of the original teacher model are 107.0 and 183.0 and the performance of the student model only decreases 11.7 and 17.3 of MAE and MSE. Based on above experimental results, it is concluded that our DKD framework is robust and can be applied to the knowledge distillation of teacher and student models with different architectures and loss functions.

TABLE IV

PERFORMANCE COMPARISON WITH THE DIFFERENT MODEL COMPRESSION ALGORITHMS ON UCF-QNRF DATASET

Methods		Params	FLOPs	MAE	MSE
Quantization	DoreFa [66]	1.35	6.9	151.3	222.4
	N2UQ [67]	21.49	10.2	138.5	186.5
Pruning	AMC [68]	5.82	27.8	113.7	200.0
	BOCR [69]	3.35	22.5	100.8	153.7
Distillation	FitNets [44]	1.35	154.7	158.5	246.1
	AT [45]			93.3	154.8
	AB [46]			103.1	159.0
	FT [70]			149.4	218.1
	PKD [71]			130.6	196.9
	CAT-KD [72]			135.6	207.4
	Our DKD			91.7	151.5

C. Comparison With Model Compression Methods

In recent years there have been many excellent studies in the field of model compression. We compare our method with the quantization, pruning, and distillation methods to verify the effectiveness of DKD. In Table IV we present the performance of the different compression algorithms on the UCF-QNRF dataset. The DoreFa and N2UQ quantize parameters of the VGG19 model, obtaining the MAE of 151.3 and 222.4 and MSE of 138.5 and 186.5, respectively. The AMC and BOCR prune the channel of convolutional neural networks automatically based on a deep-reinforcement learning algorithm and a Bayesian optimization algorithm. Among the six knowledge-distillation methods, although CAT-KD is a newly advanced algorithm with high interpretability by transferring activation maps, the AT and AB perform better than CAT-KD especially the AT method, by considering the global feature difference between the teacher and the student, achieving the performance with MAE and MSE of 93.3 and 154.8. There are only 1.6 and 4.7 of MAE and MSE weaker than our method. The AT transfers the activation-based attention map and the gradient-based attention map, and the AB method distills the activation boundary from the teacher model to the student model based on the normalization distance, such as the l1 norm in the corresponding feature maps, respectively. Our DKD distills the knowledge hierarchically from the projector to the intermediate feature maps based on OTD, which is designed for the counting task according to the critical characteristic of

TABLE V
PERFORMANCE OF KD AND DKD UNDER DIFFERENT CPRS ON THE FOUR DATASETS. PARAMS DENOTE THE NUMBER OF PARAMETERS. THE UNITS ARE MILLION (M) FOR PARAMS AND GIGA (G) FOR FLOPS

Params	CPR	SHHA (576×864)			SHHB (768×1024)			UCF_CC_50 (2101× 2888)			UCF-QNRF (2032×2912)		
		MAE	MSE	FLOPs	MAE	MSE	FLOPs	MAE	MSE	FLOPs	MAE	MSE	FLOPs
5.38	1/2	63.9 **62.2**	102.6 **102.3**	51.5	8.3 **7.5**	13.0 **12.2**	81.4	203.2 **173.8**	281.2 **242.0**	627.2	93.3 **89.3**	157.3 **153.3**	612.7
2.37	1/3	64.7 **62.4**	101.9 **99.7**	22.7	8.7 **7.6**	13.0 **13.3**	35.9	212.5 **206.8**	299.7 **278.0**	276.3	89.5 **87.8**	153.1 **152.5**	269.9
1.35	1/4	67.3 **64.4**	106.0 **103.0**	13.0	8.5 **7.4**	13.4 **12.7**	20.6	226.5 **210.3**	318.4 **283.8**	158.3	94.0 **91.7**	155.8 **150.1**	154.7
0.85	1/5	69.1 **66.4**	108.1 **105.6**	8.1	17.6 **15.2**	30.2 **21.9**	12.9	238.8 **226.9**	320.8 **300.3**	99.1	94.0 **91.9**	160.6 **155.6**	96.8
21.50	1	61.5	103.2	205.1	7.5	12.6	324.1	229.3	308.2	2496.5	87.7	158.1	2438.7

TABLE VI
PERFORMANCE COMPARISONS WITH OTHER OBJECT COUNTING MODELS ON FSC-147 DATASET

Method	Params	FLOPs	Val Set		Test Set	
			MAE	MSE	MAE	MSE
Mean	-	-	53.4	124.5	47.6	147.7
Median	-	-	48.7	129.7	47.7	152.5
Pre.GMN [73]	19.7	316.1	60.6	137.8	62.7	159.7
FR [74]	66.3	519.9	45.5	112.5	41.6	141.0
FamNet [60]	17.8	330.3	23.8	69.1	22.1	99.5
Our KD	4.5	86.1	45.6	108.9	45.5	136.1
Our DKD	**4.5**	**86.1**	**45.0**	**99.7**	**45.0**	**130.9**

point annotation. This method continuously enhances the final density map to the previous feature map, allowing the student model to fastly learn more prominent target features. Some methods such as FitNets, FT and PKD have a worse effect on the counting task. Therefore, our proposed method shows excellence in the field of crowd counting.

D. Comparison With Other Object Counting Models

To further demonstrate the robustness of the proposed DKD method, we apply the DKD method to other object counting models on the FSC-147 dataset. The FamNet [60] is a few-shot adaption and matching network that is used as the teacher model and the original model. The Params and FLOPs of FamNet are 17.8M and 330.3G and the Params and FLOPs of the student model are 4.5M and 86.1G when CPR is 1/2. The experimental results are shown in Table VI. The Mean method and Median method refer to always output the average and median object count for training images. Other compared models such as Pre.GMN meaning Pre-trained GMN [73] and FR [74] are object detectors. It can be observed there is a gap between our DKD and FamNet on the Val Set and Test Set. The MAE and MSE of DKD are 45.0 and 99.7 on Val Set, which are 21.2 and 30.6 weaker than the performance of FamNet. Compared with other models, our DKD shows better counting performance with fewer Params and FLOPs. For example, the FR model has large Params and FLOPs but performs weaker than DKD, especially in the evaluation of MSE. Therefore, our DKD can potentially apply other object counting datasets and models to efficiently obtain better counting accuracy.

E. Ablation Study

1) Channel Preservation Ratio: In this paper, we compress the counting model by reducing the channel number in every layer. In general, for models with the same network structure, the size and computation of the model are positively correlated with the counting performance. In this section, we conduct a study to evaluate the influence of CPR on the counting performance of the model.

As shown in Table V, we present the performance of BL model trained with different CPRs. To compute the FLOPs, we use the average resolution of the four datasets. The parameter amount of the original teacher model is 21.50M. We compress the model with the CPR of 1/2, 1/3, 1/4, and 1/5. When the number of parameters is only 0.85M, which is only 3.97% percentage of the original model, the performance is not degraded very much. In particular, after performing SKD, the performance of the model is greatly improved. As for the SHHA dataset, its MAE and MSE are 66.4 and 105.6, with FLOPs of 8.1G, which are only lower than 4.9 and 2.4 of the original model. The other datasets also have better performance with the lightweight model, including the small model size and the low computation. For the SHHB dataset, when CPR is 1/4, the performance of KD is 8.5 and 13.4, and the SKD is 7.5 and 12.7, which is almost the same as the original model with the MAE and MSE of 7.4 and 12.6. The Params and FLOPs of the student model are 1.35M and 20.6G, while those of the teacher model are 21.50M and 324.1G. The tiny UCF_CC_50 dataset has better performance since the performance with only 0.85M of Params is stronger than original model. We also observe that the average resolution of the image has much more influence on the FLOPs. The UCF-QNRF dataset has a high resolution of 2032×2912, and the FLOPs of the original model is 2438.7G, which really causes inference latency. After the student model is trained based on our proposed DKD framework with the Params of 2.37M and FLOPs of 269.9G, the performance is better than the original model with the MAE and MSE of 87.8 and 152.5. Based on the results of the four datasets with the different CPRs, it is clear that our method has the great ability of

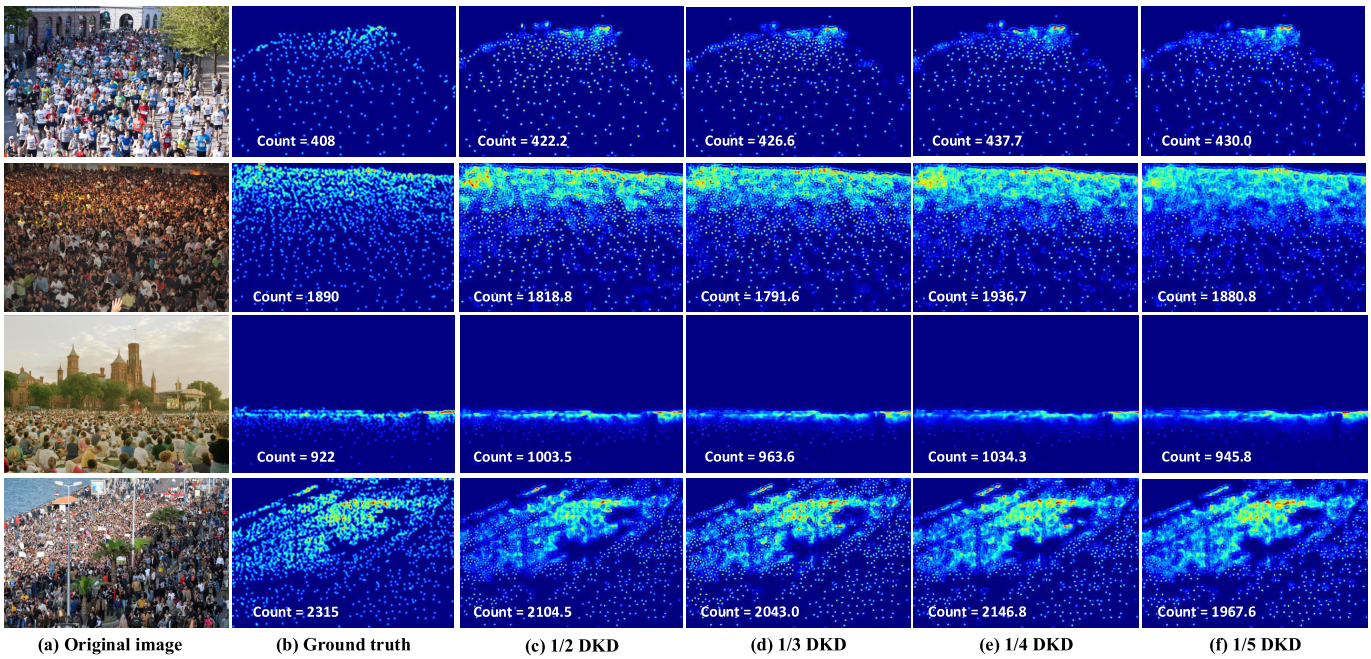

Fig. 6. Visualization of density maps on UCF-QNRF dataset generated by the four BL models with different CPRs. (a) Original image. (b) GT. (c) Estimated map by the 1/2 BL model. (d) Estimated map by the 1/3 BL model. (e) Estimated map by the 1/4 BL model. (f) Estimated map by the 1/5 BL model.

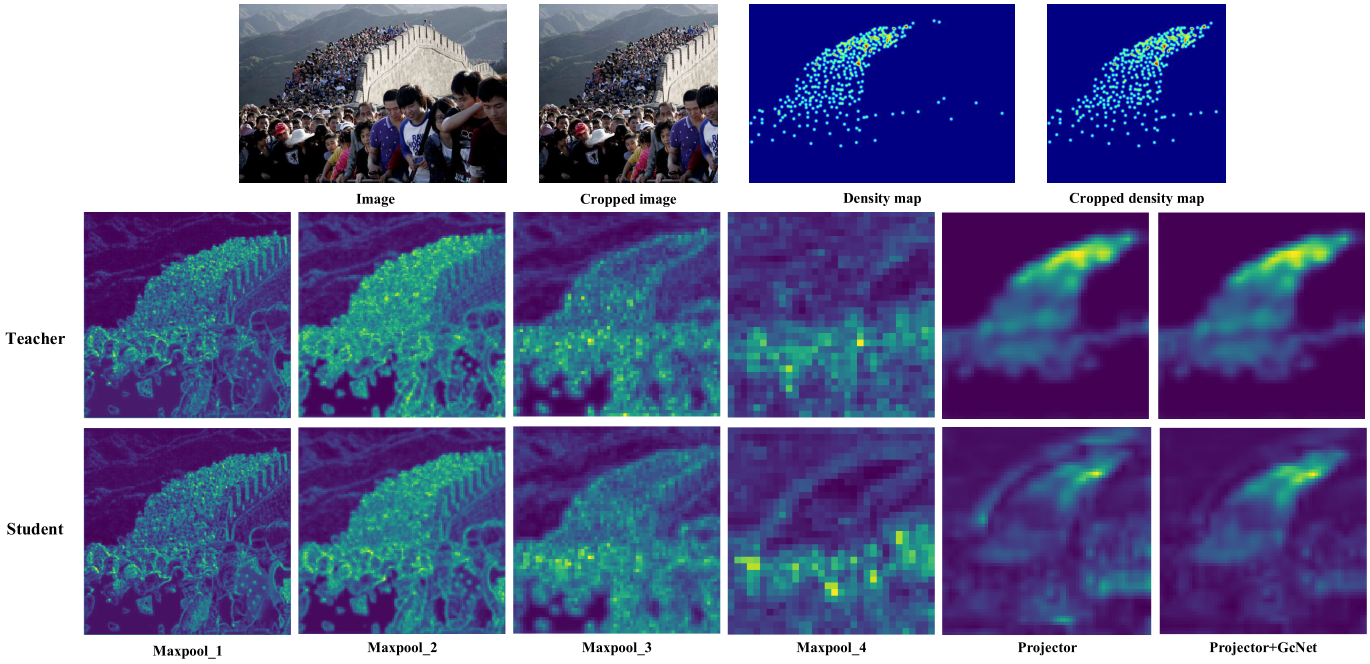

Fig. 7. Visualization of the feature maps of the teacher model and the student model on the UCF-QNRF dataset. The first row is the original image, the cropped image, the density map, and the cropped map. The second and third rows are the feature maps of the teacher model and the student model, including four MaxPooling layers, the projector, and the projector extracted by GcNet.

KD, enabling it to obtain an efficient model. In addition, the smaller models achieve even better performance than the original model, indicating that the heavy models really have redundant parameters to deal with the different datasets. It inspires us to compress the big model to obtain a more efficient student model to deploy on the edge and intelligent terminals.

Fig. 6 shows the density maps estimated by the four BL models with the different CPRs, including 1/2, 1/3, 1/4, and 1/5. The four models can all obtain the summary crowd distributions in the four images. Although the 1/5 KD only has the 4.0% parameters and FLOPs of the original model, in the second image its predicted counting error is almost less than ten people compared with the ground truth, which is an excellent result in the congested scenario. The other models also performed better in these scenarios. It is proven that our DKD model generates high-quality and accurate counts with low-latency and high-performance.

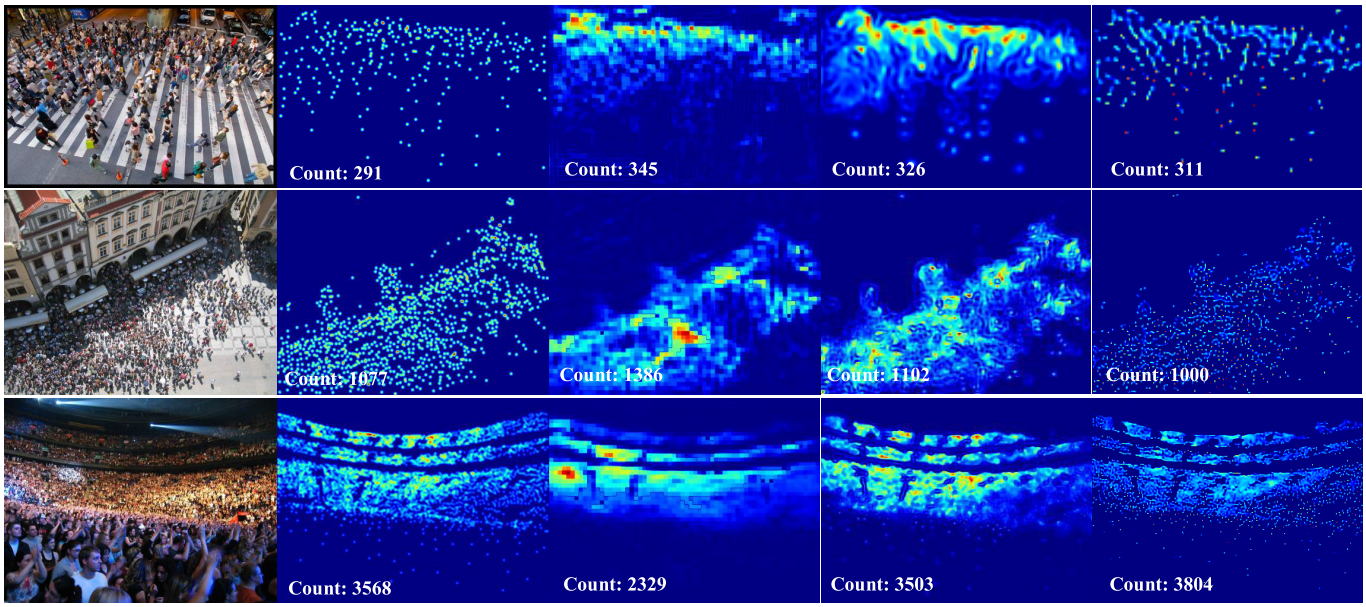

Fig. 8. Visualization of density maps on UCF-QNRF dataset. The density maps are generated by (b) GT; (c) VGG19 model; (d) BL model; (e) DMC model.

TABLE VII

PERFORMANCE OF STUDENT MODEL TRAINED WITH DIFFERENT TRANSFER CONFIGURATIONS ON UCF-QNRF DATASET IN THE KD STAGE. W/O DENOTES REMOVING THE LOSS FUNCTION

Configuration	MAE	MSE
only $\mathcal{L}_{hard}$	135.6	224.7
w/o $\mathcal{L}_{soft}$	94.9	158.2
w/o $\mathcal{L}_{cos}$	97.5	159.5
w/o $\mathcal{L}_{proj\_ot}$	95.8	156.1
w/o $\mathcal{L}_{inter\_ot}$	166.1	256.1
Our KD $\mathcal{L}_{\mathbf{total1}}$	**94.0**	**155.8**

TABLE VIII

PERFORMANCE OF STUDENT SELF-DISTILLED WITH DIFFERENT CONFIGURATIONS ON THE UCF-QNRF DATASET IN THE SKD STAGE

Configuration	MAE	MSE
w/o $\mathcal{L}_{total2}$	94.0	155.8
$\mathcal{L}_{hard}$	92.2	153.5
$\mathcal{L}_{hard}$ + $\mathcal{L}_{inter\_mse}$	88.8	152.3
$\mathcal{L}_{hard}$ + $\mathcal{L}_{proj\_mse}$	89.5	150.5
Our SKD $\mathcal{L}_{\mathbf{total2}}$	**91.7**	**150.1**

2) Knowledge-Distillation Module: To demonstrate the effectiveness of the knowledge transfer from teacher to student, we conduct experiments to evaluate the effect of transfer configurations, as shown in Table VII. There are five parts in the loss function during the knowledge transfer, and except for $\mathcal{L}_{hard}$, we evaluate the effectiveness of the other four parts. When the $\mathcal{L}_{inter\_ot}$ is removed, the performance degrades greatly with 166.1 of MAE and 256.1 of MSE, which is even worse than the performance using only the $\mathcal{L}_{hard}$. It is concluded that when the student model is initialized, the guidance of the teacher in the shallow feature maps is critical. We observe that the model without $\mathcal{L}_{soft}$ performs better, which only decreases 0.9 and 2.4 of MAE and MSE with the model trained on $\mathcal{L}_{total1}$. A possible reason is that the knowledge from previous feature maps is fully transferred from the teacher to the student, and there is little influence on the final density map. Furthermore, the $\mathcal{L}_{cos}$ and $\mathcal{L}_{proj\_ot}$ are equally important. Especially, if the $\mathcal{L}_{cos}$ is removed, the GcNet attached to the teacher model cannot be updated according to the counting performance, leading to random guidance for the student model.

We visualize the feature maps of the teacher model and the student model in the transferred layers, as shown in Fig. 7. The shallower layers have the ability to extract texture features in the context and people, and as the number of network layers increases, the edge features of the context gradually disappear. The teacher model has a sensitive perception of crowd characteristics compared with the student model. The GcNet has a more obvious impact on the student model, which eliminates some salient regions without crowd distribution in the feature map generated by the student model. In summary, our proposed scheme is effective at distilling accurate knowledge from the teacher to the student.

3) Self-Knowledge Distillation Module: To demonstrate the effectiveness of SKD, we conduct experiments to evaluate the effect of the different losses. There are three configurations that only use $\mathcal{L}_{hard}$, $\mathcal{L}_{hard}$ combined with $\mathcal{L}_{inter\_mse}$, and $\mathcal{L}_{hard}$ combined with the $\mathcal{L}_{proj\_mse}$. The experimental results are shown in Table VIII. As long as the SKD module is added, the performance of the model is improved compared with the first KD stage. When only the $\mathcal{L}_{hard}$ is used for the SKD, the performance improvement of the model is not obvious. When the $\mathcal{L}_{inter\_mse}$ or $\mathcal{L}_{proj\_mse}$ is added, the MAE is reduced by nearly four to five points, and the MSE is reduced by three to five points. When the SKD modules of the $\mathcal{L}_{inter\_mse}$ and the $\mathcal{L}_{proj\_mse}$ are arbitrarily selected, their MAE is even better than that of using both modules simultaneously. Therefore, the

TABLE IX
PERFORMANCE OF COMPLETE MODELS WITHOUT PRUNING CHANNEL BASED ON THE SKD METHOD ON UCF-QNRF DATASET

Method	MAE	MSE	Params	FLOPs
VGG19 [64]	205.0	329.8	21.5	2438.7
SKD + VGG19	**195.1**	**306.2**		
BL [18]	87.7	158.1		
SKD + BL	**85.1**	**148.1**		
DMC [31]	85.6	148.3		
SKD + DMC	**84.1**	**147.6**		

experimental results fully demonstrate the effectiveness of the SKD module, which corrects errors and bias introduced by the teacher to a certain extent, and further enhances the knowledge learning ability of the student.

4) Self-Distillation for Complete Model: To further illustrate the effect of the proposed SKD, we employ three counting models with different optimization supervisions to distill the knowledge by themselves. The VGG19, BL model, and DMC model are used to conduct SKD and they are not compressed to keep the original parameters. The BL model and DMC model both have the same backbone based on VGG19, but they adapt different loss functions to update the network in the teacher model. The parameters and the FLOPs are exactly the same for the three models. The VGG19 only uses the MSE loss function, the BL considers the contribution probability of each head and the DMC introduces the OTD to update the network. The results of the SKD are shown in Table IX. The performances of all three models improved a lot via the SKD module. It is concluded that the SKD stage in DKD shows the great potential to refine and understand the knowledge and eliminate the disturbance caused by teacher model. The VGG19 is weaker than the two other models, and the MAE and MSE of the BL model improved by 2.6 and 10.0 after implementing SKD. The MAE and MSE of the self-distilled DMC model are 84.1 and 147.6. As shown in Fig. 8, there are visualizations of the density maps generated by the VGG19 model, the BL model, and the DMC model performed by SKD. It can be observed that the density maps generated by the VGG19 model are a little rough and fuzzy, and the BL model and DMC model present clear crowd distributions and make accurate estimations. The DMC model generates more fine-grained head positions than the BL model. In summary, the heavy model based on the SKD mechanism further improves the counting performance.

V. CONCLUSION AND FUTURE WORK

In this paper, we present a novel knowledge distillation framework DKD, designed to formulate a lightweight model for crowd counting. Since the inaccurate estimation generated by the teacher model in some cases misleads the training of the student model, the proposed DKD executes KD in two stages, including knowledge transfer from the teacher to the student and self-distillation to eliminate the estimation errors. The adaptive perspective KD and OTD are applied to obtain more comprehensive information from the local and global views of the feature maps. This greatly enhances the ability of the feature extraction to improve the quality of the density map generated by the student model. Extensive experiments on four popular crowd counting datasets demonstrate that our DKD framework outperforms state-of-the-art lightweight approaches with fewer parameters and FLOPs. It shows the great potential of the model compression task in the field of crowd counting. We expect the lightweight models to meet the demands of low-latency and high-performance in real applications for the scenario of crowd management.

The DKD is effective in the field of crowd counting, while it also shows the superior performance in other counting task. The idea of DKD can be considered to apply to other common tasks through the adaptive modifications. In addition, the limitation of DKD is that, despite improving the parameter efficiency of CNN, it is urgent to perform KD on the transformer-based counting models to obtain efficient models. In future work, We will continue to further explore the generalized KD framework to improve the lightweight level of large deep counting models.

REFERENCES

[1] Z. Du, M. Shi, J. Deng, and S. Zafeiriou, "Redesigning multi-scale neural network for crowd counting," *IEEE Trans. Image Process.*, vol. 32, pp. 3664–3678, 2023.
[2] H. Lin, Z. Ma, R. Ji, Y. Wang, and X. Hong, "Boosting crowd counting via multifaceted attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 19628–19637.
[3] J. Wan, N. S. Kumar, and A. B. Chan, "Fine-grained crowd counting," *IEEE Trans. Image Process.*, vol. 30, pp. 2114–2126, 2021.
[4] J. Cheng, H. Xiong, Z. Cao, and H. Lu, "Decoupled two-stage crowd counting and beyond," *IEEE Trans. Image Process.*, vol. 30, pp. 2862–2875, 2021.
[5] D. Liang, J. Xie, Z. Zou, X. Ye, W. Xu, and X. Bai, "CrowdCLIP: Unsupervised crowd counting via vision-language model," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 2893–2903.
[6] Z.-Q. Cheng, Q. Dai, H. Li, J. Song, X. Wu, and A. G. Hauptmann, "Rethinking spatial invariance of convolutional networks for object counting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 19606–19616.
[7] Y. Tian, Y. Lei, J. Zhang, and J. Z. Wang, "PaDNet: Pan-density crowd counting," *IEEE Trans. Image Process.*, vol. 29, pp. 2714–2727, 2020.
[8] Y. Liu et al., "Crowd counting via cross-stage refinement networks," *IEEE Trans. Image Process.*, vol. 29, pp. 6800–6812, 2020.
[9] Y. Li, X. Zhang, and D. Chen, "CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1091–1100.
[10] Y.-J. Ma, H.-H. Shuai, and W.-H. Cheng, "Spatiotemporal dilated convolution with uncertain matching for video-based crowd estimation," *IEEE Trans. Multimedia*, vol. 24, pp. 261–273, 2022.
[11] X. Cao, Z. Wang, Y. Zhao, and F. Su, "Scale aggregation network for accurate and efficient crowd counting," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 734–750.
[12] D. Liang, X. Chen, W. Xu, Y. Zhou, and X. Bai, "TransCrowd: Weakly-supervised crowd counting with transformers," *Sci. China Inf. Sci.*, vol. 65, no. 6, pp. 1–14, Jun. 2022.
[13] J. Gao, M. Gong, and X. Li, "Congested crowd instance localization with dilated convolutional Swin Transformer," *Neurocomputing*, vol. 513, pp. 94–103, Nov. 2022.
[14] J. Gao, Q. Wang, and X. Li, "PCC Net: Perspective crowd counting via spatial convolutional network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 10, pp. 3486–3498, Oct. 2020.
[15] Z. Dong, R. Zhang, X. Shao, and Y. Li, "Scale-recursive network with point supervision for crowd scene analysis," *Neurocomputing*, vol. 384, pp. 314–324, Apr. 2020.
[16] Y. Hao, J. Wang, D. Huo, N. Guizani, L. Hu, and M. Chen, "Digital twin-assisted URLLC-enabled task offloading in mobile edge network via robust combinatorial optimization," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 10, pp. 3022–3033, Oct. 2023.

[17] L. Liu, J. Chen, H. Wu, T. Chen, G. Li, and L. Lin, "Efficient crowd counting via structured knowledge transfer," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 2645–2654.

[18] Z. Ma, X. Wei, X. Hong, and Y. Gong, "Bayesian loss for crowd count estimation with point supervision," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6141–6150.

[19] Z. Tian et al., "Adaptive perspective distillation for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 2, pp. 1372–1387, Feb. 2023.

[20] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "GCNet: Non-local networks meet squeeze-excitation networks and beyond," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 1971–1980.

[21] C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman, "Interactive object counting," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 504–518.

[22] Z. Ma and A. B. Chan, "Crossing the line: Crowd counting by integer programming with local features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2539–2546.

[23] V. A. Sindagi and V. M. Patel, "HA-CCN: Hierarchical attention-based crowd counting network," *IEEE Trans. Image Process.*, vol. 29, pp. 323–335, 2020.

[24] R. Wang, Y. Hao, L. Hu, J. Chen, M. Chen, and D. Wu, "Self-supervised learning with data-efficient supervised fine-tuning for crowd counting," *IEEE Trans. Multimedia*, vol. 25, pp. 1538–1546, 2023.

[25] M. Rodriguez, I. Laptev, J. Sivic, and J.-Y. Audibert, "Density-aware person detection and tracking in crowds," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2423–2430.

[26] Y. Liu, Z. Wang, M. Shi, S. Satoh, Q. Zhao, and H. Yang, "Towards unsupervised crowd counting via regression-detection bi-knowledge transfer," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 129–137.

[27] W. Ren, X. Wang, J. Tian, Y. Tang, and A. B. Chan, "Tracking-by-counting: Using network flows on crowd density maps for tracking multiple targets," *IEEE Trans. Image Process.*, vol. 30, pp. 1439–1452, 2021.

[28] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 23, 2010, pp. 1–9.

[29] Y. Yang, G. Li, D. Du, Q. Huang, and N. Sebe, "Embedding perspective analysis into multi-column convolutional neural network for crowd counting," *IEEE Trans. Image Process.*, vol. 30, pp. 1395–1407, 2021.

[30] D. B. Sam and R. V. Babu, "Top-down feedback for crowd counting convolutional neural network," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 7323–7330.

[31] B. Wang, H. Liu, D. Samaras, and M. H. Nguyen, "Distribution matching for crowd counting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 1595–1607.

[32] Q. Song et al., "Rethinking counting and localization in crowds: A purely point-based framework," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 3345–3354.

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

[34] Z. Ma, X. Wei, X. Hong, and Y. Gong, "Learning scales from points: A scale-aware probabilistic model for crowd counting," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 220–228.

[35] M. Jiang, J. Lin, and Z. J. Wang, "ShuffleCount: Task-specific knowledge distillation for crowd counting," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2021, pp. 999–1003.

[36] Z. Duan, S. Wang, H. Di, and J. Deng, "Distillation remote sensing object counting via multi-scale context feature aggregation," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–12, 2021.

[37] X. Ma, S. Du, and Y. Liu, "A lightweight neural network for crowd analysis of images with congested scenes," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 979–983.

[38] X. Shi, X. Li, C. Wu, S. Kong, J. Yang, and L. He, "A real-time deep network for crowd counting," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 2328–2332.

[39] X. Jiang et al., "Crowd counting and density estimation by trellis encoder–decoder networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6133–6142.

[40] P. Wang, C. Gao, Y. Wang, H. Li, and Y. Gao, "MobileCount: An efficient encoder–decoder framework for real-time crowd counting," *Neurocomputing*, vol. 407, pp. 292–299, Sep. 2020.

[41] Y. Liu, G. Cao, H. Shi, and Y. Hu, "Lw-Count: An effective lightweight encoding-decoding crowd counting network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 10, pp. 6821–6834, Oct. 2022.

[42] Y. Wang, Z. Ma, X. Wei, S. Zheng, Y. Wang, and X. Hong, "ECCNAS: Efficient crowd counting neural architecture search," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 18, no. 1, pp. 1–19, Feb. 2022.

[43] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[44] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "FitNets: Hints for thin deep nets," 2014, *arXiv:1412.6550*.

[45] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," 2016, *arXiv:1612.03928*.

[46] B. Heo, M. Lee, S. Yun, and J. Y. Choi, "Knowledge transfer via distillation of activation boundaries formed by hidden neurons," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, vol. 33, no. 1, 2019, pp. 3779–3787.

[47] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4133–4141.

[48] N. Passalis, M. Tzelepi, and A. Tefas, "Probabilistic knowledge transfer for lightweight deep representation learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 5, pp. 2030–2039, May 2021.

[49] T. He, C. Shen, Z. Tian, D. Gong, C. Sun, and Y. Yan, "Knowledge adaptation for efficient semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 578–587.

[50] J. Guo et al., "Distilling object detectors via decoupled features," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2154–2164.

[51] L. Zhang, J. Song, A. Gao, J. Chen, C. Bao, and K. Ma, "Be your own teacher: Improve the performance of convolutional neural networks via self distillation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3713–3722.

[52] L. Zhang, C. Bao, and K. Ma, "Self-distillation: Towards efficient and compact neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 8, pp. 4388–4403, Aug. 2022.

[53] K. Kim, B. Ji, D. Yoon, and S. Hwang, "Self-knowledge distillation with progressive refinement of targets," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6567–6576.

[54] L. Chen et al., "Adversarial text generation via feature-mover's distance," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–12.

[55] S. N. Afriat, "Theory of maxima and the method of Lagrange," *SIAM J. Appl. Math.*, vol. 20, no. 3, pp. 343–357, May 1971.

[56] J. Liu et al., "Discrimination-aware network pruning for deep model compression," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 8, pp. 4035–4051, Aug. 2022.

[57] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 589–597.

[58] H. Idrees, I. Saleemi, C. Seibert, and M. Shah, "Multi-source multi-scale counting in extremely dense crowd images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2547–2554.

[59] H. Idrees et al., "Composition loss for counting, density map estimation and localization in dense crowds," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 544–559.

[60] V. Ranjan, U. Sharma, T. Nguyen, and M. Hoai, "Learning to count everything," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 3394–3403.

[61] W. Liu, M. Salzmann, and P. Fua, "Context-aware crowd counting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5099–5108.

[62] Q. Wang, J. Gao, W. Lin, and Y. Yuan, "Learning from synthetic data for crowd counting in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8198–8207.

[63] D. B. Sam, S. V. Peri, M. N. Sundararaman, A. Kamath, and R. V. Babu, "Locate, size, and count: Accurately resolving people in dense crowds via detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 8, pp. 2739–2751, Aug. 2021.

[64] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[65] R. Wang et al., "AAC: Automatic augmentation for crowd counting," *Neurocomputing*, vol. 500, pp. 90–98, Aug. 2022.

[66] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, and Y. Zou, "DoReFa-Net: Training low bitwidth convolutional neural networks with low bitwidth gradients," 2016, *arXiv:1606.06160*.

[67] Z. Liu, K.-T. Cheng, D. Huang, E. Xing, and Z. Shen, "Nonuniform-to-uniform quantization: Towards accurate quantization via generalized straight-through estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 4942–4952.
[68] Y. He, J. Lin, Z. Liu, H. Wang, L.-J. Li, and S. Han, "AMC: AutoML for model compression and acceleration on mobile devices," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 815–832.
[69] H. Fan, J. Mu, and W. Zhang, "Bayesian optimization with clustering and rollback for CNN auto pruning," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 494–511.
[70] J. Kim, S. Park, and N. Kwak, "Paraphrasing complex network: Network compression via factor transfer," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–10.
[71] W. Cao, Y. Zhang, J. Gao, A. Cheng, K. Cheng, and J. Cheng, "PKD: General distillation framework for object detectors via Pearson Correlation Coefficient," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 15394–15406.
[72] Z. Guo, H. Yan, H. Li, and X. Lin, "Class attention transfer based knowledge distillation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 11868–11877.
[73] E. Lu, W. Xie, and A. Zisserman, "Class-agnostic counting," in *Proc. 14th Asian Conf. Comput. Vis.*, Perth, WA, Australia. Cham, Switzerland: Springer, 2019, pp. 669–684.
[74] B. Kang, Z. Liu, X. Wang, F. Yu, J. Feng, and T. Darrell, "Few-shot object detection via feature reweighting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8420–8429.

Rui Wang received the bachelor's degree in computer science and technology from Lanzhou University, China, in 2018. She is currently pursuing the Ph.D. degree with the Embedded and Pervasive Computing (EPIC) Laboratory, School of Computer Science and Technology, Huazhong University of Science and Technology, China. Her research interests include computer vision, emotion recognition, and edge computing.

Yixue Hao (Member, IEEE) received the Ph.D. degree in computer science from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2017. He is currently an Associate Professor with the School of Computer Science and Technology, HUST. His current research interests include 5G networks, the Internet of Things, edge computing, edge caching, and cognitive computing.

Long Hu is currently an Associate Professor with the School of Computer Science and Technology, Huazhong University of Science and Technology (HUST), China. He was a Visiting Student with the Department of Electrical and Computer Engineering, The University of British Columbia, from August 2015 to April 2017. His research includes the Internet of Things, software defined networking, caching, 5G, body area networks, body sensor networks, and mobile cloud computing.

Xianzhi Li received the Ph.D. degree from the Department of Computer Science and Engineering, The Chinese University of Hong Kong. She is currently an Associate Professor with the Huazhong University of Science and Technology. Prior to that, she was a Postdoctoral Fellow with The Chinese University of Hong Kong. She serves as a reviewer of several conferences and journals, including IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, CVPR, and ICCV. Her research interests include 3D vision, computer graphics, and deep learning.

Min Chen (Fellow, IEEE) has been a Full Professor with the School of Computer Science and Engineering, South China University of Technology. He is also the Director of the Embedded and Pervasive Computing (EPIC) Laboratory, Huazhong University of Science and Technology (HUST). He is the founding Chair of IEEE Computer Society Special Technical Communities on Big Data. He was an Assistant Professor with the School of Computer Science and Engineering, Seoul National University, before he joined HUST. He is the Chair of IEEE GLOBECOM 2022 eHealth Symposium. His Google Scholar Citations reached 43,000+ with an H-index of 97. His top paper was cited more than 4,090 times. He was selected as a Highly Cited Researcher from 2018 to 2022. He got IEEE Communications Society Fred W. Ellersick Prize in 2017, the IEEE Jack Neubauer Memorial Award in 2019, and IEEE ComSoc APB Outstanding Paper Award in 2022. He is a fellow of IET.

Yiming Miao (Member, IEEE) received the Ph.D. degree from the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China, in 2021. She is currently a Research Assistant Professor with the School of Data Science, The Chinese University of Hong Kong, Shenzhen, China. She is a reviewer of IEEE WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, *IEEE Network* magazine, *FGCS*, *JCSC*, and *ACM CSUR*. She was the Symposium Chair for IWCMC 2022 and 2021 and the Web Chair for TRIDENTCOM 2017 and the CloudComp 2016. Her research interests include edge computing, 5G mobile communication system, the Internet of Things, unmanned aerial vehicle, blockchain, and wireless sensor networks.

Iztok Humar (Senior Member, IEEE) received the Ph.D. degrees in telecommunications from the Faculty of Electrical Engineering (FE) and in information management from the Faculty of Economics, University of Ljubljana, Slovenia, in 2007 and 2009, respectively. He is currently a Full Professor with the Faculty of Electrical Engineering (FE), University of Ljubljana, where he lecturers on design, management, and modeling of telecommunication networks. His main research interests include the design, planning and management of communications networks and services, and edge cognitive computing and the modeling of networks and traffic for energy efficiency and QoS/QoE. He served as the IEEE Communication Society of Slovenia Chapter Chair and the IEEE Slovenia Section Secretary.