Efficient Crowd Counting via Structured Knowledge Transfer

Lingbo Liu
Sun Yat-Sen University
liulingb@mail2.sysu.edu.cn

Jiaqi Chen
Sun Yat-Sen University
jadgechen@gmail.com

Hefeng Wu*
Sun Yat-Sen University
wuhefeng@gmail.com

Tianshui Chen
DarkMatter AI Research
tianshuichen@gmail.com

Guanbin Li
Sun Yat-Sen University
liguanbin@mail.sysu.edu.cn

Liang Lin
Sun Yat-Sen University
DarkMatter AI Research
linliang@ieee.org

ABSTRACT

Crowd counting is an application-oriented task and its inference efficiency is crucial for real-world applications. However, most previous works relied on heavy backbone networks and required prohibitive run-time consumption, which would seriously restrict their deployment scopes and cause poor scalability. To liberate these crowd counting models, we propose a novel Structured Knowledge Transfer (SKT) framework, which fully exploits the structured knowledge of a well-trained teacher network to generate a lightweight but still highly effective student network. Specifically, it is integrated with two complementary transfer modules, including an Intra-Layer Pattern Transfer which sequentially distills the knowledge embedded in layer-wise features of the teacher network to guide feature learning of the student network and an Inter-Layer Relation Transfer which densely distills the cross-layer correlation knowledge of the teacher to regularize the student's feature evolution. Consequently, our student network can derive the layer-wise and cross-layer knowledge from the teacher network to learn compact yet effective features. Extensive evaluations on three benchmarks well demonstrate the effectiveness of our SKT for extensive crowd counting models. In particular, only using around 6% of the parameters and computation cost of original models, our distilled VGG-based models obtain at least 6.5× speed-up on an Nvidia 1080 GPU and even achieve state-of-the-art performance. Our code and models are available at https://github.com/HCPLab-SYSU/SKT.

CCS CONCEPTS

• **Computing methodologies** → *Machine learning*; • **Applied computing** → *Surveillance mechanisms*.

KEYWORDS

crowd counting; knowledge transfer; network compression and acceleration

---

*Hefeng Wu is the corresponding author. Lingbo Liu and Jiaqi Chen are co-first authors.

1 INTRODUCTION

Crowd counting, whose objective is to automatically estimate the total number of people in surveillance scenes, is an important issue of crowd analysis [27, 70]. With the rapid growth of urban population and the increasing demand for security analysis and early warning in large-scale crowded scenarios, this task has attracted extensive interest in academic and industrial fields, due to its wide-ranging applications in video surveillance [74], congestion alerting [50] and traffic prediction [33, 34].

Recently, deep neural networks [4, 6, 10, 32, 35, 41, 44, 58, 68, 71, 72, 76] have become mainstream in the task of crowd counting and have made remarkable progress. To acquire better performance, most of the state-of-the-art methods [13, 28, 31, 36, 40, 62, 66] utilized heavy backbone networks (such as the VGG model [56]) to extract features. Nevertheless, requiring large computation cost and running at low speeds, these models are exceedingly inefficient. As shown in Table 1, the latest DISSNet [31] costs 3.7 seconds on an Nvidia 1080 GPU and 379 seconds on an Intel Xeon CPU to process an input image of size 2032×2912. This would seriously restrict their deployment scopes and cause poor scalability, particularly on edge computing devices [1] with limited computing resources. Moreover, to handle citywide surveillance videos in real-time, we may need thousands of high-performance GPUs, which are expensive and energy-consuming. Under these circumstances, a cost-effective model is extremely desired for crowd counting.

Thus, one fundamental question is that *how we can acquire an efficient crowd counting model from existing well-trained but heavy networks.* A series of efforts [5, 8, 14, 37, 61] have been made to compress and speed-up deep neural networks. However, most of them either require cumbersome hyper-parameters search (e.g., the sensitivity in per layer for parameters pruning) or rely on specific hardware platforms (e.g., weight quantization and half-precision floating-point computing). Recently, knowledge distillation [19], which trains a small student network to acquire the knowledge of a complex teacher network, has become a desirable alternative for model compression due to its broad applicability areas. Numerous works [9, 42, 46, 49, 69, 75] have verified its effectiveness for image

Method	RMSE	#Param	FLOPs	GPU	CPU
DISSNet [31]	159.20	8.86	8670.09	3677.98	378.80
CAN [36]	183.00	18.10	2594.18	972.16	149.56
CSRNet* [28]	233.32	16.26	2447.91	823.84	119.67
BL* [40]	158.09	21.50	2441.23	595.72	130.76
Ours	**156.82**	**1.35**	**155.30**	**90.96**	**9.78**

Table 1: The Root Mean Squared Error (RMSE), Parameters, FLOPs, and inference time of our SKT network and four state-of-the-art models on the UCF-QNRF [22] dataset. The FLOPs and parameters are computed with the input size of 2032×2912, and the inference times are measured on a Intel Xeon E5 CPU (2.4G) and a single Nvidia GTX 1080 GPU. The RMSE of the models with * are reimplemented by us. The units are million (M) for #Param, giga (G) for FLOPs, millisecond (ms) for GPU time, and second (s) for CPU time, respectively. More efficiency analysis can be found in Table 8.

classification. However, it is difficult to apply knowledge distillation to the more challenging dense-labeling crowd counting, since this task requires the distilled models to maintain the discriminative abilities at every location. In this case, extensive knowledge is desired to be transferred to well equip the abilities of student networks for generating accurate crowd density maps.

To fully distill the knowledge of a pre-trained teacher network, we conduct a more in-depth analysis of the crowd counting models and observe that the structured knowledge of deep networks is implicitly embedded in **i)** *layer-wise features* that carry image content, and **ii)** *cross-layer correlations* that encode feature updating schemas. In this work, we develop a novel framework termed Structured Knowledge Transfer (SKT), which simultaneously distills the knowledge of layer-wise features and cross-layer correlation via two complementary transfer modules, i.e., an Intra-Layer Pattern Transfer (Intra-PT) and an Inter-Layer Relation Transfer (Inter-RT). **First**, our Intra-PT takes a set of representative features extracted from a well-trained teacher network to sequentially supervise the corresponding features of a student network, analogous to using the teacher's knowledge to progressively correct the student's learning deviation. As a result, the features of the student network exhibit similar visual or semantic patterns of its supervisor. **Second**, our Intra-PT densely computes the relationships between pairwise features of the teacher network and then utilizes such knowledge to help the student network regularize the long short-term evolution of its hierarchical features. Thereby, the student network can learn the solution procedure flow of its teacher. Thanks to the tailor-designed SKT framework, our lightweight student network can effectively learn compact and knowledgeable features, yielding high-quality crowd density maps.

In experiments, we apply the proposed SKT framework to compress and accelerate a series of existing crowd counting models (e.g, CSRNet [28], BL [40] and SANet [6]). Extensive evaluations on three representative benchmarks greatly demonstrate the effectiveness of our method. Under the condition of occupying only nearly 6% of the original model parameters and computational cost, our distilled VGG-based models obtain at least 6.5× speed-up on GPU and 9× speed-up on CPU. Moreover, these lightweight models can preserve competitive performance, and even achieve state-of-the-art

results on Shanghaitech[76] Part-A and UCF-QNRF [22] datasets. In summary, the major contributions of this work are three-fold:

- We propose a general and comprehensive Structured Knowledge Transfer framework, which can generate lightweight but effective crowd counting models. To the best of our knowledge, we are the first to focus on improving the efficiency of crowd counting models.
- Two cooperative knowledge transfers (i.e., an Intra-Layer Pattern one and an Inter-Layer Relation one) are incorporated to fully distill the structured knowledge of well-trained models to lightweight models for crowd counting.
- Extensive experiments on three benchmarks show the effectiveness of our method. In particular, our distilled VGG-based models have an order of magnitude speed-up while achieving state-of-the-art performance.

2 RELATED WORKS

Crowd Counting: Crowd counting has been extensively studied for decades. Early works [11, 26] estimated the crowd count by directly locating the people with pedestrian detectors. Subsequently, some methods [7, 47] learned a mapping between handcrafted features and crowd count with regressors. Only using image low-level information, these methods had high efficiencies, but their performance was far from satisfactory for real-world applications.

Recently, we have witnessed the great success of convolutional neural networks [29, 32, 43, 48, 53, 55, 59, 63, 66, 71, 72] in crowd counting. Most of these previous approaches focused on how to improve the performance of deep models. To this end, they tended to use heavy backbone networks (e.g., the VGG model [56]) to extract representative features. For instance, Li et al. [28] combined a VGG-16 based front-end network and dilated convolutional back-end network to learn hierarchical features for crowd counting. Liu et al. [36] introduced an expanded context-aware network that learned both image features and geometry features with two truncated VGG16 models. Liu et al. [31] utilized three paralleled VGG16 networks to extract multiscale features and then conducted structured refinements. Recently, Ma et al. [40] proposed a new Bayesian loss for crowd counting and verified its effectiveness on VGG19. Although the aforementioned methods can make impressive progress, their performance advantages come with the cost of burdensome computation. Thus it is hard to directly apply these methods to practical applications. In contrast, we take into consideration both the performance and computation cost. In this work, we aim to improve the efficiency of existing crowd counting models under the condition of preserving the performance.

Model Compression: Parameters quantization [12], parameters pruning [14] and knowledge distillation [19] are three types of commonly-used algorithms for model compression. Specifically, quantization methods [23, 77] compress networks by reducing the number of bits required to represent weights, but they usually rely on specific hardware platforms. Pruning methods [17, 25, 78] removed redundant weights or channels of layers. However, most of them used weight masks to simulate the pruning and mass post-processing are needed to achieve real speed-up. By contrast, knowledge distillation [42, 49, 75] is more general and its objective is to transfer knowledge from a heavy network to a lightweight one. Recently, it has been widely studied. For instance, Hinton et al. [19]
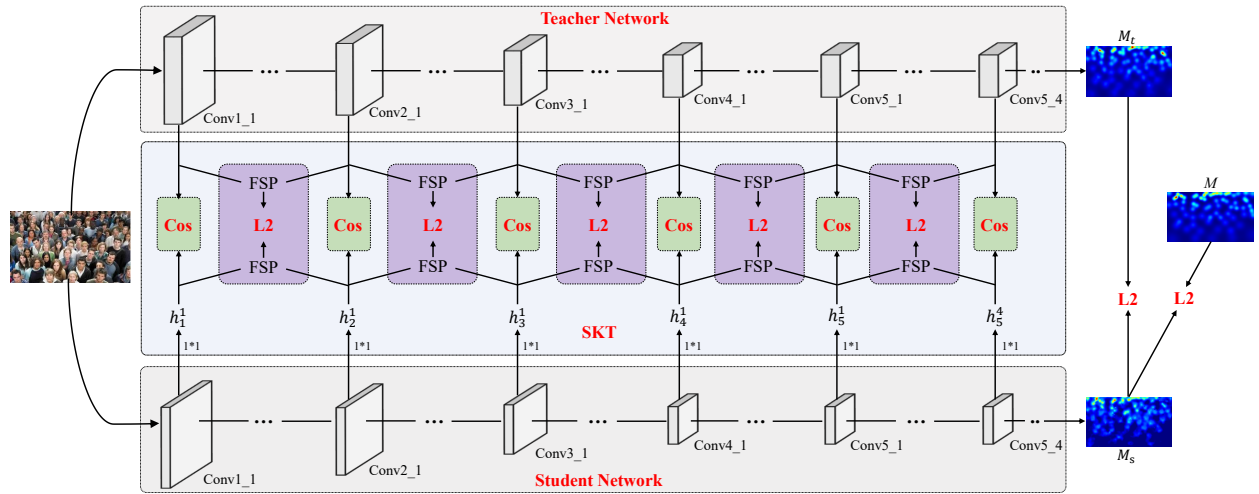
Figure 1: The proposed Structured Knowledge Transfer (SKT) framework for crowd counting. With two complementary distillation modules, our SKT can effectively distill the structured knowledge of a pre-trained teacher network to a small student network. First, an Intra-Layer Pattern Transfer sequentially distill the inherent knowledge in a teacher's feature to enhance a student's feature with a *cosine* metric. Second, an Inter-Layer Relation Transfer enforces the student network to learn the long short term feature relationships of the teacher network, thereby fully mimicking the flow of solution procedure (FSP) of teacher. Notice that FSP matrices are densely computed between some representative features in our framework. For the conciseness and beautification of this figure, we only show some FSP matrices of adjacent features.

trained a distilled network with the soft output of a large highly regularized network. Romero et al. [46] improved the performance of student networks with both the outputs and the intermediate features of teacher networks. Zagoruyko and Komodakis [69] utilized activation-based and gradient-based spatial attention maps to transfer knowledge between two networks. Nevertheless, most of these previous methods were proposed for image classification. Recently, for fast human pose estimation, Zhang et al. [73] directly used the final pose maps of teacher networks to supervise the interim and final results of student networks. [16] and [39] distilled the knowledge embedded in layer-wise features for semantic segmentation, but both of them neglected the the cross-layer correlation knowledge. In contrast, our method fully explores the structured knowledge (e.g., layer-wise one and and cross-layer one) of teacher networks to optimize student networks for compact and effective feature learning.

3 METHOD

In this work, a general Structured Knowledge Transfer (SKT) framework is proposed to address the efficiency problem of existing crowd counting models. Its architecture is shown in Fig. 1. Specifically, an Intra-Layer Pattern Transfer (Intra-PT) and an Inter-Layer Relation Transfer (Inter-RT) are incorporated into our framework to fully transfer the structured knowledge (e.g., layer-wise one and cross-layer one) from the teacher networks to the student networks.

In this section, we take the VGG16-based CSRNet [28] as an example to introduce the working modules of our SKT framework. The student network is a $1/n$-CSRNet, in which the channel number of each convolutional layer (but except the last layer) is $1/n$ of the original one in CSRNet. Compared with the heavy CSRNet, the lightweight $1/n$-CSRNet model only has $1/n^2$ parameters and

computation cost, but it suffers serious performance degradation. Thus, our objective is to improve the performance of $1/n$-CSRNet as far as possible by transferring the knowledge of CSRNet. Notice that our SKT is general and it is also applicable to other crowd counting models (e.g., BL [40] and SANet [6]). Several distilled models are analyzed and compared in Section 4.

3.1 Feature Extraction

In general, teacher networks have been pre-trained on standard benchmarks. The learned knowledge can be explicitly represented as parameters or implicitly embedded into features. Similar to the previous works [46, 69], we perform knowledge transfer on the feature level. The knowledge in the features of teacher networks is treated as supervisory information and can be utilized to guide the representation learning of student networks. Therefore, before conducting knowledge transfer, we need to extract the hierarchical features of teacher/student networks in advance.

As shown in Fig. 1, given an unconstrained image I, we simultaneously feed it into CSRNet and $1/n$-CSRNet for feature extraction. For convenience, the j-th dilated convolutional layer in the backend network of CSRNet is renamed as "Conv5_j" in our work. Thus the feature at layer Conv$i$_j of CSRNet can be uniformly denoted as t_i^j. Similarly, we use s_i^j to represent the Conv$i$_j feature of $1/n$-CSRNet. Notice that t_i^j and s_i^j share the same resolution, but s_i^j has only $1/n$ channels of t_i^j. Since the Inter-RT in Section 3.3 computes feature relations densely, we only perform distillation on some representative features, in order to reduce the computational cost during the training phase. Specifically, the selected features are shown as follows:

$$T = \{t_1^1,\ t_2^1,\ t_3^1,\ t_4^1,\ t_5^1,\ t_5^4\}, \quad S = \{s_1^1,\ s_2^1,\ s_3^1,\ s_4^1,\ s_5^1,\ s_5^4\}, \quad (1)$$

where T and S are the feature groups of CSRNet and $1/n$-CSRNet, respectively.

3.2 Intra-Layer Pattern Transfer

As described in the above subsection, the extracted feature t_i^j implicitly contains the learned knowledge of CSRNet. To improve the performance of the lightweight $1/n$-CSRNet, we design a simple but effective Intra-Layer Pattern Transfer (Intra-PT) module, which sequentially transfers the knowledge of selected features of CSRNet to the corresponding features of $1/n$-CSRNet. Formally, we enforce s_i^j to learn the visual/semantic patterns of t_i^j and optimize the parameters of $1/n$-CSRNet by maximizing their distribution similarity.

Specifically, our Intra-PT is composed of two steps. The **first** step is channel adjustment. As feature s_i^j and t_i^j have different channel numbers, it is unsuitable to directly compute their similarity. To eliminate this issue, we generate a group of interim features $H = \{h_1^1, \ h_2^1, \ h_3^1, \ h_4^1, \ h_5^1, \ h_5^4\}$ by feeding each feature s_i^j in S into a 1×1 convolutional layer, which is expressed as:

$$h_i^j = s_i^j * w_{1\times1}, \qquad (2)$$

where $w_{1\times1}$ denotes the parameters of the convolutional layer. The output h_i^j is the embedding feature of s_i^j and its channel number is the same as that of t_i^j.

The **second** step is similarity computation and knowledge transfer. Since Euclidean distance is too restrictive and may cause the rote learning of student networks, we adopt a relatively liberal metric, *cosine*, to measure the similarity of two features. Specifically, the similarity of t_i^j and h_i^j at location (x, y) is calculated by:

$$
\begin{aligned}
\mathcal{S}_i^j(x, y) &= Cos\{t_i^j(x, y), h_i^j(x, y)\}, \\
&= \sum_{c=1}^{C_i^j} \frac{t_i^j(x, y, c) \cdot h_i^j(x, y, c)}{|t_i^j(x, y)| \cdot |h_i^j(x, y)|},
\end{aligned}
\qquad (3)
$$

where C_i^j denotes the channel number of feature t_i^j and $t_i^j(x, y, c)$ is the response value of t_i^j at location (x, y) of the c-th channel. The symbol $|*|$ is the length of a vector. Thus, the loss function of our Intra-PTD is defined as follows:

$$\mathcal{L}_{Intra} = \sum_{t_i^j, h_i^j} \sum_{x=1}^{\mathcal{H}_i^j} \sum_{y=1}^{\mathcal{W}_i^j} 1 - \mathcal{S}_i^j(x, y), \qquad (4)$$

where $\mathcal{H}_i^j$ and $\mathcal{W}_i^j$ are the height and width of feature t_i^j. By minimizing this simple loss and back-propagating the gradients, our method can effectively transfer knowledge and optimize the parameters of $1/n$-CSRNet. Compared with the pair-wise distillation [39] which is heavy computed and can only conducted on high-layer features with low resolutions, our Intra-PT is more efficient and can fully transfer the knowledge embedded at various layers.

3.3 Inter-Layer Relation Transfer

"Teaching one to fish is better than giving him fish". Thus, a student network should also be encouraged to learn how to solve a problem. Inspired by [67], the flow of solution procedure (FSP) can be modeled with the relationship between features from two layers. Such a relationship is a kind of meaningful knowledge. In this subsection, we develop an Inter-Layer Relation Transfer (Inter-RT) module, which densely computes the pairwise feature relationships (FSP matrices) of the teacher network to regularize the long short-term feature evolution of the student network .

Let's introduce the detail of our Inter-RT. We first present the generation of FSP matrix. For two general feature $f_1 \in R^{h \times w \times m}$ and $f_2 \in R^{h \times w \times n}$, we compute their FSP matrix $\mathcal{F}(f_1, f_2) \in R^{m \times n}$ with channel-wise inner product. Specifically, its value at index (c_1, c_2) is calculated by:

$$\mathcal{F}_{c_1, c_2}(f_1, f_2) = \sum_{x=1}^{h} \sum_{y=1}^{w} \frac{f_1(x, y, c_1) \cdot f_2(x, y, c_2)}{h \cdot w}. \qquad (5)$$

Notice that FSP matrix computation is conducted on features with same resolution. However, the features in T have various resolutions. To address this issue and simultaneously reduce the FSP computation cost, we consistently resize all features in group T to the resolution of t_5^4 with max pooling. The resized feature of t_i^j is denoted as $R(t_i^j)$. In the same way, all features in group H are also resized to the resolution of h_5^4.

Rather than only compute FSP matrices for adjacent features, we design a Dense FSP strategy to better capture the long-short term evolution of features. Specifically, we generate a FSP matrix $\mathcal{F}\{R(t_i^j), R(t_k^l)\}$ for every pair of features (t_i^j, t_k^l) in T. Similarly, a matrix $\mathcal{F}\{R(h_i^j), R(h_k^l)\}$ is also computed for every pair of features (h_i^j, h_k^l) in H. Finally, the loss function of our Inter-RT is calculated as follows:

$$\mathcal{L}_{Inter} = \sum_{t_i^j, h_i^j} \sum_{t_k^l, h_k^l} ||\mathcal{F}\{R(t_i^j), R(t_k^l)\} - \mathcal{F}\{R(h_i^j), R(h_k^l)\}||^2. \quad (6)$$

By minimizing the distances of these FSP matrices, the knowledge of CSRNet can be transferred to $1/n$-CSRNet.

3.4 Learn from Soft Ground-Truth

In our work, a density map generated from point annotations is termed as hard ground-truth. We find that the density maps predicted by the teacher network are complementary to hard ground-truths. As shown in Fig. 2, there may exist some blemishes (e.g., the inaccurate scales and positions of human heads, the unmarked heads) in some regions of hard ground-truths. Fortunately, with powerful knowledge, a well-trained teacher network may predict some relatively reasonable maps. These predicted density maps can also be treated as knowledge and we call them soft ground-truths. In this work, we train our student network with both the hard and soft ground-truths.

As shown in Fig. 1, we use M to represent the hard ground-truth of image I. The predicted map of CSRNet is denoted as M_t and the output map of $1/n$-CSRNet is denoted as M_s. Since $1/n$-CSRNet is expected to simultaneously learn the knowledge of hard ground-truth and soft ground-truth, we defined the loss function on density maps as follows:

$$\mathcal{L}_m = ||M_s - M||^2 + ||M_s - M_t||^2. \qquad (7)$$

(a) Image   (b) Hard Ground-Truth

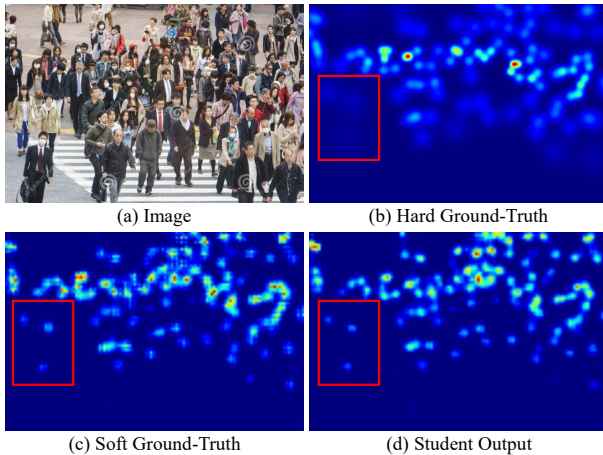(c) Soft Ground-Truth   (d) Student Output

Figure 2: Illustration the complementarity of hard and soft ground-truth (GT). (b) is a hard GT generated from point annotations with geometry-adaptive Gaussian kernels. (c) is a soft GT predicted by CSRNet, while (d) is the estimated map of $1/n$-CSRNet. The hard GT may exist some blemishes (e.g., the inaccurate scales and positions of human heads, the unmarked heads). For example, the red box in (b) shows the human heads with inaccurate scales. We find that the soft GT may be relatively reasonable in some regions and it is complementary to the hard GT. Thus, they can be incorporated to train the student network.

Finally, we optimize the parameters of 1/n-CSRNet by minimizing the total losses of all knowledge transfers:

$$\mathcal{L} = \alpha_1 \cdot L_{Intra} + \alpha_2 \cdot \mathcal{L}_{Inter} + \alpha_3 \cdot \mathcal{L}_m, \quad (8)$$

where α_1, α_2 and α_3 are weights of different losses.

4 EXPERIMENTS

4.1 Experiment Settings

In this work, we conduct extensive experiments on the following three public benchmarks for crowd counting.

Shanghaitech [76]: This dataset contains 1,198 images with 330,165 annotated individuals. It is composed of two parts: Part-A contains 482 images of congested crowd scenes, where 300 images are used for training and 182 for testing, and Part-B contains 716 images of sparse crowd scenes, with 400 images for training and the rest for testing.

UCF-QNRF [22]: As one of the most challenging datasets, UCF-QNRF contains 1,535 images captured from unconstrained crowd scenes with huge variations in scale, density and viewpoint. Specifically, 1,201 images are used for training and 334 for testing. There are about 1.25 million annotated people in this dataset and the number of persons per image varies from 49 to 12,865.

WorldExpo'10 [72]: It contains 1,132 surveillance videos captured by 108 cameras during the Shanghai WorldExpo 2010. Specifically, 3,380 images from 103 scenes are used as the training set and 600 images from other five scenes as the test set. Region-of-Interest (ROI) are provided to specify the counting regions for the test set.

CPR	#Param	FLOPs	RMSE	Transfer
1	16.26	205.88	105.99	
1/2	4.07	51.77	137.32	
			113.61	✓
1/3	1.81	23.11	140.29	
			114.68	✓
1/4	1.02	13.09	146.40	
			114.40	✓
1/5	0.64	8.45	149.40	
			118.78	✓

Table 2: Performance under different channel preservation rates (CPR) on Shanghaitech Part-A. #Param denotes the number of parameters (M). FLOPs is the number of FLoating point OPerations (G) and it is computed on a 576×864 image (576×864 is the average resolution on Shanghaitech Part-A).

Following [6, 28], we adopt Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) to quantitatively evaluate the performance of crowd counting. Specifically, they are defined as follows:

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} \|P_i - G_i\|, \text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \|P_i - G_i\|^2}, \quad (9)$$

where N is the number of test images, P_i and G_i are the predicted and ground-truth count of ith image, respectively.

4.2 Ablation Study

4.2.1 **Exploration on Channel Preservation Rate.** In this work, we compress the existing crowd counting models by reducing their channel numbers. A model can run more efficiently if it has fewer parameters/channels, however, at the expense of performance. Thus, the balance between efficiency and accuracy should be investigated. In this section, we first conduct ablation study to evaluate the influence of Channel Preservation Rate (CPR) on the performance of models.

In Table 2, we summarize the performance of various CSRNet trained with different CPRs. As can be observed, the original CSRNet has 16.26M parameters and achieves an RMSE of 105.99 using 205.88G FLOPs. When preserving half the number of channels, 1/2-CSRNet has a 4× reduction in parameters and FLOPs. However, without applying knowledge transfer, its performance seriously degrades, with the RMSE increasing to 137.32. By applying our SKT, 1/2-CSRNet exhibits an obvious performance gain. When CPR decreases to 1/4, we can observe the model is further reduced in model size and FLOP consumption with only 6.25% of the original one. What's more, the 1/4-CSRNet with knowledge transfer just has a negligible performance drop. When CPR further decreases, 1/5-CSRNet meets a relatively large performance drop with a smaller gain in parameters and FLOPs reduction. Therefore, we consider it roughly reaches a balance when CPR is 1/4 and this setting is widely adopted in the following experiments.

4.2.2 **Effect of Different Transfer Configurations.** We further perform experiments to evaluate the effect of different transfer configurations of our framework. This ablation study is conducted
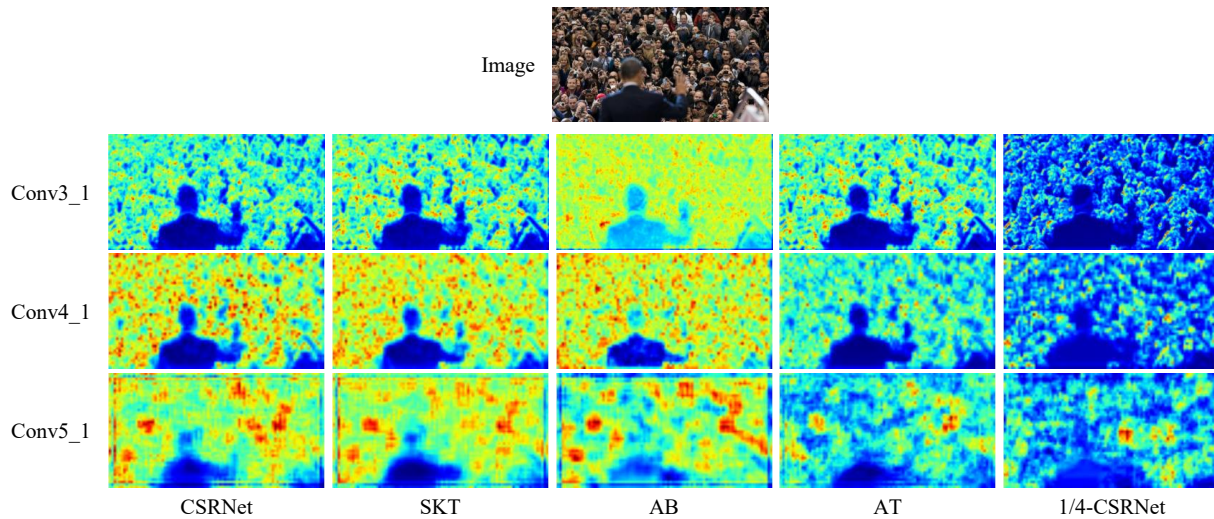
Figure 3: Visualization of the feature maps of different models on Shanghaitech Part-A. The first and fifth columns are the features of the complete CSRNet and the naive 1/4-CSRNet. The middle three columns show the student features of 1/4-CSRNet+SKT, 1/4-CSRNet+AB [18] and 1/4-CSRNet+AT [69]. The bottom three rows are the channel-wise average features at layers Conv3_1, Conv4_1 and Conv5_1, respectively. Thanks to the tailor-designed Intra-PT and Inter-RT, our 1/4-CSRNet+SKT can fully absorb the structured knowledge of CSRNet, thus the generated features are very similar to those of the teacher network.

Transfer Configuration		MAE	RMSE
W/O Transfer		89.65	146.40
Intra-PT	L2	76.61	120.56
	Cos	74.99	117.58
Inter-RT	S-FSP	79.22	133.21
	D-FSP	73.25	120.77
Intra-PT & Inter-RT	L2 + D-FSP	72.89	117.92
	Cos + D-FSP	71.55	114.40

Table 3: Performance of 1/4-CSRNet distilled with different transfer configurations on Shanghaitech Part-A. D-FSP and S-FSP refer to Dense FSP and Sparse FSP, respectively.

Ground-Truth Type	MAE	RMSE
Hard	72.94	116.68
Soft	74.89	118.33
Hard + Soft	71.55	114.40

Table 4: Performance of 1/4-CSRNet trained with different ground-truths on Shanghaitech Part-A.

Method		MAE	RMSE
Baseline	CSRNet	68.43	105.99
	1/4-CSRNet	89.65	146.40
Quantization	DoReFa [77]	80.02	124.1
	QAT [23]	75.50	128.09
Pruning	L1Filter [25]	85.18	135.82
	CP [17]	82.05	130.65
	AGP [78]	78.51	125.83
Distillation	FitNets [46]	87.32	140.34
	DML [75]	85.23	138.10
	NST [20]	76.26	116.57
	AT [69]	74.65	127.06
	AB [18]	75.73	123.28
	SKT (Ours)	**71.55**	**114.40**

Table 5: Performance of different compression algorithms on Shanghaitech Part-A.

based on 1/4-CSRNet and the results are summarized in Table 3. When just trained with our Intra-Layer Pattern Transfer module, the 1/4-CSRNet is able to obtain evident performance gain, decreasing MAE by at least 13 and RMSE by at least 25.8. We observe that using the Cosine (Cos) as the similarity metric performs better than using Euclidian (L2) distance. The possible reason is that the Cosine metric enforces the consistency of feature distribution between teacher and student networks, while the Euclidean distance further enforces location-wise similarity which is too restrictive for knowledge transfer. On the other hand, only using our Inter-Layer

Relation Transfer module can also boost the 1/4-CSRNet's performance by a large margin, with MAE decreased by at least 10 and RMSE decreased by at least 13. It is worth noting that the Dense FSP strategy achieves quite impressive performance gain, decreasing MAE of the 1/4-CSRNet without transfer by 16.40 (relatively 18.3%). When combining both the proposed Intra-Layer and Intra-Layer transfers to form our overall framework, the 1/4-CSRNet's performance is further boosted. Specifically, by using Cosine metric and Dense FSP, the 1/4-CSRNet achieves the best performance (MAE 71.55, RMSE 114.40) within all transfer configurations of our framework.

4.2.3 **Effect of Soft Ground-Truth.** In this section, we conduct experiments to evaluate the effect of soft ground-truth (GT) on the

Method	Part-A		Part-B	
	MAE	RMSE	MAE	RMSE
MCNN [76]	110.2	173.2	26.4	41.3
SwitchCNN [48]	90.4	135	21.6	33.4
DecideNet [30]	-	-	21.5	31.9
CP-CNN [58]	73.6	106.4	20.1	30.1
DNCL [55]	73.5	112.3	18.7	26.0
ACSCP [52]	75.7	102.7	17.2	27.4
L2R [38]	73.6	112.0	13.7	21.4
IG-CNN [2]	72.5	118.2	13.6	21.1
IC-CNN [45]	68.5	116.2	10.7	16.0
CFF [54]	65.2	109.4	7.2	12.2
SANet*	75.33	122.2	10.45	17.92
1/4-SANet	97.36	155.43	14.79	23.43
1/4-SANet+SKT	**78.02**	**126.58**	**11.86**	**19.83**
CSRNet*	68.43	105.99	7.49	12.33
1/4-CSRNet	89.65	146.40	10.82	16.21
1/4-CSRNet + SKT	**71.55**	**114.40**	**7.48**	**11.68**
BL*	61.46	103.17	7.50	12.60
1/4-BL	88.35	145.47	12.25	19.77
1/4-BL + SKT	**62.73**	**102.33**	**7.98**	**13.13**

Table 6: Performance comparison on Shanghaitech dataset. The models with symbol * are our reimplemented teacher networks.

Method	MAE	RMSE
Idrees et al. [21]	315	508
MCNN [76]	277	426
Encoder-Decoder [3]	270	478
CMTL [57]	252	514
SwitchCNN [48]	228	445
Resnet-101 [15]	190	277
CL [22]	132	191
TEDnet [24]	113	188
CAN [36]	107	183
S-DCNet [65]	104.40	176.10
DSSINet [31]	99.10	159.20
SANet*	152.59	246.98
1/4-SANet	192.47	293.96
1/4-SANet + SKT	**157.46**	**257.66**
CSRNet*	145.54	233.32
1/4-CSRNet	186.31	287.65
1/4-CSRNet + SKT	**144.36**	**234.64**
BL*	87.70	158.09
1/4-BL	135.64	224.72
1/4-BL + SKT	**96.24**	**156.82**

Table 7: Performance of different methods on UCF-QNRF dataset. The models with * are our reimplemented teacher networks.

performance. As shown in Table 4, when only using the soft GT generated by the teacher network as supervision, the 1/4-CSRNet's performance is slightly worse than that of using hard GT. But it also indicates that the soft GT does provide useful information since it does not cause severe performance degradation. Furthermore, it is witnessed that the model's performance is promoted when we utilize both soft GT and hard GT to supervise model training. This further demonstrates that the soft GT is complementary to the hard GT and we can indeed transfer knowledge of the teacher network with soft GT.

4.3 Comparison with Model Compression Algorithms

Undoubtedly, some existing compression algorithms can also be applied to compress the crowd counting models. To verify the superiority of the proposed SKT, we compare our method with ten representative compression algorithms.

In Table 5, we summarize the performance of different compression algorithms on Shanghaitech Part-A. Specifically, quantizing the parameters of CSRNet with 8 bits, DoReFa [77] and QAT [23] obtain an MAE of 80.02/75.50 respectively. When we employ the official setting of CP [17] to prune CSRNet, the compressed model obtains an MAE of 82.05 with 6.89M parameters. To maintain the same number of parameters of 1/4-CSRNet, L1Filter [25] and AGP [78] prunes 93.75% parameters, and their MAE are above 78. Furthermore, six distillation methods including our SKT are applied to distill CSRNet to 1/4-CSRNet. As can be observed, our method achieves the best performance w.r.t both MAE and RMSE. The feature visualization in Fig. 3 also shows that our features are much better than those of other compression methods. These quantitative and qualitative superiorities are attributed to that the tailor-designed Intra-PT and

Inter-RT can fully distill the knowledge of the teacher networks. What's more, the proposed SKT is easily implemented and the distilled crowd counting models can be directly deployed on various edge devices. In summary, our SKT fits most with the crowd counting task, among the various existing compression algorithms.

4.4 Comparison with Crowd Counting Methods

To demonstrate the effectiveness of the proposed SKT, we also conduct comparisons with state-of-the-art methods of crowd counting from both performance and efficiency perspectives. Besides CSRNet [28], we also apply our SKT framework to distill other two representative models BL [40] and SANet [6]. Specifically, the former is based on VGG19 and we obtain a lightweight 1/4-BL with the same transfer configuration of CSRNet. Similar to GoogLeNet [60], the latter SANet adopts multi-column blocks to extract features. For SANet, we transfer knowledge on the output features of each block, yielding a lightweight 1/4-SANet.

4.4.1 **Performance Comparison.** The performance comparison with recent state-of-the-art methods on Shanghaitech, UCF-QNRF and WorldExpo'10 datasets are reported in Tables 6, 7 and 9, respectively. As can be observed, the BL model is the existing best-performing method, achieving the lowest MAE and RMSE on almost all these datasets. The CSRNet and SANet also show a relatively good performance among the compared methods. However, when reduced in model size to gain efficiency, the 1/4-BL, 1/4-CSRNet and 1/4-SANet models without knowledge transfer have a heavy performance degradation, compared with the original models. By applying our SKT method, these lightweight models can obtain comparable results with the original models, and even achieve better performance on some datasets. For example, as shown in Table 6,

Method	#Param	Shanghaitech A (576×864)			Shanghaitech B (768×1024)			WorldExpo'10 (576×720)			UCF-QNRF (2032×2912)		
		FLOPs	GPU	CPU	FLOPs	GPU	CPU	FLOPs	GPU	CPU	FLOPs	GPU	CPU
DSSINet [31]	8.86	729.20	296.32	32.39	1152.31	471.83	49.49	607.66	250.69	26.13	8670.09	3677.98	378.80
CAN [36]	18.10	218.20	79.02	7.99	344.80	117.12	20.75	181.83	68.00	6.84	2594.18	972.16	149.56
CSRNet [28]	16.26	205.88	66.58	7.85	325.34	98.68	19.17	171.57	57.57	6.51	2447.91	823.84	119.67
BL [40]	21.50	205.32	47.89	8.84	324.46	70.18	19.63	171.10	40.52	6.69	2441.23	595.72	130.76
SANet [6]	0.91	33.55	35.20	3.90	52.96	52.85	11.42	27.97	29.84	3.13	397.50	636.48	87.50
1/4-CSRNet + SKT	1.02	13.09	8.88	0.87	20.69	12.65	1.84	10.91	7.71	0.67	155.69	106.08	9.71
1/4-BL + SKT	1.35	13.06	7.40	0.88	20.64	10.42	1.89	10.88	6.25	0.69	155.30	90.96	9.78
1/4-SANet + SKT	0.058	2.52	11.83	1.10	3.98	16.86	2.10	2.10	9.72	0.92	29.92	368.04	18.64

Table 8: The inference efficiency of state-of-the-art methods. #Param denotes the number of parameters, while FLOPs is the number of FLoating point OPerations. The execution time is computed on an Nvidia GTX 1080 GPU and a 2.4 GHz Intel Xeon E5 CPU. The units are million (M) for #Param, giga (G) for FLOPs, millisecond (ms) for GPU time, and second (s) for CPU time.

Method	S1	S2	S3	S4	S5	Avg
Zhang et al [72]	9.8	14.1	14.3	22.2	3.7	12.9
MCNN [76]	3.4	20.6	12.9	13.0	8.1	11.6
Shang et al. [51]	7.8	15.4	14.9	11.8	5.8	11.7
IG-CNN [2]	2.6	16.1	10.1	20.2	7.6	11.3
ConvLSTM [64]	7.1	15.2	15.2	13.9	3.5	10.9
IC-CNN [45]	17.0	12.3	9.2	8.1	4.7	10.3
SwitchCNN [48]	4.4	15.7	10.0	11.0	5.9	9.4
DecideNet [30]	2.00	13.14	8.90	17.40	4.75	9.23
DNCL [55]	1.9	12.1	20.7	8.3	2.6	9.1
CP-CNN [58]	2.9	14.7	10.5	10.4	5.8	8.86
PGCNet [66]	2.5	12.7	8.4	13.7	3.2	8.1
TEDnet [24]	2.3	10.1	11.3	13.8	2.6	8.0
SANet*	2.92	15.22	14.86	14.73	4.20	10.39
1/4-SANet	3.77	19.93	19.33	18.42	6.36	13.56
1/4-SANet + SKT	**3.42**	**16.13**	**15.82**	**15.37**	**4.91**	**11.13**
CSRNet*	1.58	13.55	14.70	7.29	3.28	8.08
1/4-CSRNet	1.96	15.70	20.59	8.52	3.70	10.09
1/4-CSRNet + SKT	**1.77**	**12.32**	**14.49**	**7.87**	**3.10**	**7.91**
BL*	1.79	10.70	14.12	7.08	3.19	7.37
1/4-BL	1.97	18.39	28.95	8.12	3.94	12.27
1/4-BL + SKT	**1.41**	**10.45**	**13.10**	**7.63**	**4.08**	**7.34**

Table 9: MAE of different methods on the WorldExpo'10 dataset. The models with symbol * are our reimplemented teacher networks.

our 1/4-CSRNet+SKT performs better in both MAE and RMSE than the original CSRNet on Shanghaitech Part-B, while 1/4-BL+SKT obtains a new state-of-the-art RMSE of 102.33 on Shanghaitech Part-A. It can also be observed from Table 7 that 1/4-BL+SKT achieves an impressive state-of-the-art RMSE of 156.82 on the UCF-QNRF dataset. Such superior performance is attributed to following reasons: **i)** Our distilled models fully absorbs the structured knowledge of the teacher networks; **ii)** Having less parameters, these models can effectively alleviate the overfitting of crowd counting.

4.4.2 **Efficiency Comparison.** A critical goal of this work is to achieve model efficiency. To further verify the superiority of SKT. we also compare our method with existing crowd counting models on inference efficiency. In Table 8, we summarize the model sizes and the inference efficiencies of different models. Specifically, the inference time of using GPU or only CPU to process an image with the average resolution for each dataset are reported comprehensively, along with the number of the consumed FLOPs. The average

resolution of images on each dataset is listed in the first row of Table 8.

As can be observed, all original models except SANet have a large number of parameters. When we compress these models with the proposed SKT, the generated models have a 16× reduction in model size and FLOPs, meanwhile achieving an order of magnitude speed-up. For example, when testing a 2032×2912 image from UCF-QNRF, our 1/4-BL+SKT only requires 90.96 milliseconds on GPU and 9.78 seconds on CPU, being 6.5/13.4× faster than the original BL model. On Shanghaitech Part-A, our 1/4-CSRNet+SKT takes 13.09 milliseconds on GPU (7.5× speed-up) and 8.88 seconds on CPU (9.0× speed-up) to process a 576×864 image. Interestingly, we find that 1/4-SANet+SKT runs slower than 1/4-BL+SKT, although SANet is much faster than BL. It mainly results from that 1/4-SANet+SKT has many stacked/parallel features with small volumes, and feature communication/synchronization consumes some extra time. In summary, the distilled VGG-based models can achieve very impressive efficiencies and satisfactory performance.

5 CONCLUSION

In this work, we propose a general Structured Knowledge Transfer (SKT) framework to improve the efficiencies of existing crowd counting models. Specifically, an Intra-Layer Pattern Transfer and an Inter-Layer Relation Transfer are incorporated to fully transfer the structured knowledge (e.g., layer-wise one and and cross-layer one) from a heavy teacher network to a lightweight student network. Extensive evaluations on three standard benchmarks show that the proposed SKT can efficiently compress extensive models of crowd counting (e.g, CSRNet, BL and SANet). In particular, our distilled VGG-based models can achieve at least 6.5× speed-up on GPU and 9.0× speed-up on CPU, and meanwhile preserve very competitive performance.

REFERENCES

[1] [n.d.]. https://en.wikipedia.org/wiki/Edge_computing.
[2] Deepak Babu Sam, Neeraj N Sajjan, R Venkatesh Babu, and Mukundhan Srinivasan. 2018. Divide and Grow: Capturing Huge Diversity in Crowd Images With Incrementally Growing CNN. In *CVPR*. 3618–3626.
[3] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. 2015. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561* (2015).
[4] Lokesh Boominathan, Srinivas SS Kruthiventi, and R Venkatesh Babu. 2016. Crowdnet: A deep convolutional network for dense crowd counting. In *ACM MM*. ACM, 640–644.
[5] Zhaowei Cai, Xiaodong He, Jian Sun, and Nuno Vasconcelos. 2017. Deep learning with low precision by half-wave gaussian quantization. In *CVPR*. 5918–5926.
[6] Xinkun Cao, Zhipeng Wang, Yanyun Zhao, and Fei Su. 2018. Scale Aggregation Network for Accurate and Efficient Crowd Counting. In *ECCV*. 734–750.
[7] Ke Chen, Chen Change Loy, Shaogang Gong, and Tony Xiang. 2012. Feature Mining for Localised Crowd Counting.. In *BMVC*, Vol. 1. 3.
[8] Tianshui Chen, Liang Lin, Wangmeng Zuo, Xiaonan Luo, and Lei Zhang. 2018. Learning a wavelet-like auto-encoder to accelerate deep neural networks. In *AAAI*.
[9] Tianshui Chen, Wenxi Wu, Yuefang Gao, Le Dong, Xiaonan Luo, and Liang Lin. 2018. Fine-grained representation learning and recognition by exploiting hierarchical semantic embedding. In *Proceedings of the 26th ACM international conference on Multimedia*. 2023–2031.
[10] Zhi-Qi Cheng, Jun-Xiu Li, Qi Dai, Xiao Wu, Jun-Yan He, and Alexander G Hauptmann. 2019. Improving the learning of multi-column convolutional neural network for crowd counting. In *Proceedings of the 27th ACM International Conference on Multimedia*. 1897–1906.
[11] Weina Ge and Robert T Collins. 2009. Marked point processes for crowd counting. In *CVPR*. IEEE, 2913–2920.
[12] Yunchao Gong, Liu Liu, Ming Yang, and Lubomir Bourdev. 2014. Compressing deep convolutional networks using vector quantization. *arXiv preprint arXiv:1412.6115* (2014).
[13] Dan Guo, Kun Li, Zheng-Jun Zha, and Meng Wang. 2019. Dadnet: Dilated-attention-deformable convnet for crowd counting. In *Proceedings of the 27th ACM International Conference on Multimedia*. 1823–1832.
[14] Song Han, Huizi Mao, and William J Dally. 2015. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149* (2015).
[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. 770–778.
[16] Tong He, Chunhua Shen, Zhi Tian, Dong Gong, Changming Sun, and Youliang Yan. 2019. Knowledge adaptation for efficient semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 578–587.
[17] Yihui He, Xiangyu Zhang, and Jian Sun. 2017. Channel pruning for accelerating very deep neural networks. In *ICCV*. 1389–1397.
[18] Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. 2019. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *AAAI*, Vol. 33. 3779–3787.
[19] Geoffrey E Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the Knowledge in a Neural Network. *arXiv: Machine Learning* (2015).
[20] Zehao Huang and Naiyan Wang. 2017. Like what you like: Knowledge distill via neuron selectivity transfer. *arXiv preprint arXiv:1707.01219* (2017).
[21] Haroon Idrees, Imran Saleemi, Cody Seibert, and Mubarak Shah. 2013. Multi-source multi-scale counting in extremely dense crowd images. In *CVPR*. 2547–2554.
[22] Haroon Idrees, Muhmmad Tayyab, Kishan Athrey, Dong Zhang, Somaya Al-Maadeed, Nasir Rajpoot, and Mubarak Shah. 2018. Composition Loss for Counting, Density Map Estimation and Localization in Dense Crowds. In *ECCV*.
[23] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. 2018. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *CVPR*. 2704–2713.
[24] Xiaolong Jiang, Zehao Xiao, Baochang Zhang, Xiantong Zhen, Xianbin Cao, David Doermann, and Ling Shao. 2019. Crowd Counting and Density Estimation by Trellis Encoder-Decoder Networks. In *CVPR*. 6133–6142.
[25] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. 2016. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710* (2016).
[26] Min Li, Zhaoxiang Zhang, Kaiqi Huang, and Tieniu Tan. 2008. Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection. In *ICPR*. IEEE, 1–4.
[27] Teng Li, Huan Chang, Meng Wang, Bingbing Ni, Richang Hong, and Shuicheng Yan. 2015. Crowded Scene Analysis: A Survey. *T-CSVT* 25, 3 (2015), 367–386.
[28] Yuhong Li, Xiaofan Zhang, and Deming Chen. 2018. CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes. In *CVPR*. 1091–1100.
[29] Dongze Lian, Jing Li, Jia Zheng, Weixin Luo, and Shenghua Gao. 2019. Density Map Regression Guided Detection Network for RGB-D Crowd Counting and Localization. In *CVPR*. 1821–1830.
[30] Jiang Liu, Chenqiang Gao, Deyu Meng, and Alexander G Hauptmann. 2018. Decidenet: Counting varying density crowds through attention guided detection and density estimation. In *CVPR*. 5197–5206.
[31] Lingbo Liu, Zhilin Qiu, Guanbin Li, Shufan Liu, Wanli Ouyang, and Liang Lin. 2019. Crowd Counting with Deep Structured Scale Integration Network. In *ICCV*. 1774–1783.
[32] Lingbo Liu, Hongjun Wang, Guanbin Li, Wanli Ouyang, and Liang Lin. 2018. Crowd counting using deep recurrent spatial-aware network. In *IJCAI*.
[33] Lingbo Liu, Ruimao Zhang, Jiefeng Peng, Guanbin Li, Bowen Du, and Liang Lin. 2018. Attentive Crowd Flow Machines. In *ACM MM*. ACM, 1553–1561.
[34] Lingbo Liu, Jiajie Zhen, Guanbin Li, Geng Zhan, Zhaocheng He, Bowen Du, and Liang Lin. 2020. Dynamic Spatial-Temporal Representation Learning for Traffic Flow Prediction. *IEEE Transactions on Intelligent Transportation Systems* (2020).
[35] Ning Liu, Yongchao Long, Changqing Zou, Qun Niu, Li Pan, and Hefeng Wu. 2019. ADCrowdNet: An Attention-injective Deformable Convolutional Network for Crowd Understanding. In *CVPR*. 3225–3234.
[36] Weizhe Liu, Mathieu Salzmann, and Pascal Fua. 2019. Context-Aware Crowd Counting. In *CVPR*. 5099–5108.
[37] Xingyu Liu, Jeff Pool, Song Han, and William J Dally. 2018. Efficient sparse-winograd convolutional neural networks. *arXiv preprint arXiv:1802.06367* (2018).
[38] Xialei Liu, Joost van de Weijer, and Andrew D Bagdanov. 2018. Leveraging Unlabeled Data for Crowd Counting by Learning to Rank. In *CVPR*.
[39] Yifan Liu, Ke Chen, Chris Liu, Zengchang Qin, Zhenbo Luo, and Jingdong Wang. 2019. Structured knowledge distillation for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2604–2613.
[40] Zhiheng Ma, Xing Wei, Xiaopeng Hong, and Yihong Gong. 2019. Bayesian loss for crowd count estimation with point supervision. In *ICCV*. 6142–6151.
[41] Zhiheng Ma, Xing Wei, Xiaopeng Hong, and Yihong Gong. 2020. Learning Scales from Points: A Scale-aware Probabilistic Model for Crowd Counting. In *ACM International Conference on Multimedia*.
[42] Seyed-Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, and Hassan Ghasemzadeh. 2019. Improved knowledge distillation via teacher assistant: Bridging the gap between student and teacher. *arXiv preprint arXiv:1902.03393* (2019).
[43] Daniel Onoro-Rubio and Roberto J López-Sastre. 2016. Towards perspective-free object counting with deep learning. In *ECCV*. Springer, 615–629.
[44] Zhilin Qiu, Lingbo Liu, Guanbin Li, Qing Wang, Nong Xiao, and Liang Lin. 2019. Crowd counting via multi-view scale aggregation networks. In *ICME*. IEEE, 1498–1503.
[45] Viresh Ranjan, Hieu Le, and Minh Hoai. 2018. Iterative Crowd Counting. In *ECCV*.
[46] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2014. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550* (2014).
[47] D. Ryan, S. Denman, C. Fookes, and S. Sridharan. 2009. Crowd Counting Using Multiple Local Features. In *DICTA*. 81–88.
[48] Deepak Babu Sam, Shiv Surya, and R Venkatesh Babu. 2017. Switching convolutional neural network for crowd counting. In *CVPR*, Vol. 1. 6.
[49] Bharat Bhusan Sau and Vineeth N Balasubramanian. 2016. Deep model compression: Distilling knowledge from noisy teachers. *arXiv preprint arXiv:1610.09650* (2016).
[50] T Semertzidis, K Dimitropoulos, A Koutsia, and N Grammalidis. 2010. Video sensor network for real-time traffic monitoring and surveillance. *IET intelligent transport systems* 4, 2 (2010), 103–112.
[51] Chong Shang, Haizhou Ai, and Bo Bai. 2016. End-to-end crowd counting via joint learning local and global count. In *ICIP*. IEEE, 1215–1219.
[52] Zan Shen, Yi Xu, Bingbing Ni, Minsi Wang, Jianguo Hu, and Xiaokang Yang. 2018. Crowd Counting via Adversarial Cross-Scale Consistency Pursuit. In *CVPR*. 5245–5254.
[53] Miaojing Shi, Zhaohui Yang, Chao Xu, and Qijun Chen. 2019. Revisiting perspective information for efficient crowd counting. In *CVPR*. 7279–7288.
[54] Zenglin Shi, Pascal Mettes, and Cees GM Snoek. 2019. Counting with focus for free. In *ICCV*. 4200–4209.
[55] Zenglin Shi, Le Zhang, Yun Liu, Xiaofeng Cao, Yangdong Ye, Ming-Ming Cheng, and Guoyan Zheng. 2018. Crowd Counting With Deep Negative Correlation Learning. In *CVPR*. 5382–5390.
[56] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
[57] Vishwanath A Sindagi and Vishal M Patel. 2017. Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting. In *AVSS*. IEEE, 1–6.
[58] Vishwanath A Sindagi and Vishal M Patel. 2017. Generating high-quality crowd density maps using contextual pyramid cnns. In *ICCV*. IEEE, 1879–1888.
[59] Vishwanath A Sindagi and Vishal M Patel. 2019. Multi-Level Bottom-Top and Top-Bottom Feature Fusion for Crowd Counting. In *ICCV*. 1002–1012.

[60] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *CVPR*. 1–9.

[61] Cheng Tai, Tong Xiao, Yi Zhang, Xiaogang Wang, et al. 2015. Convolutional neural networks with low-rank regularization. *arXiv preprint arXiv:1511.06067* (2015).

[62] Xin Tan, Chun Tao, Tongwei Ren, Jinhui Tang, and Gangshan Wu. 2019. Crowd Counting via Multi-layer Regression. In *Proceedings of the 27th ACM International Conference on Multimedia*. 1907–1915.

[63] Elad Walach and Lior Wolf. 2016. Learning to count with CNN boosting. In *ECCV*. Springer, 660–676.

[64] Feng Xiong, Xingjian Shi, and Dit-Yan Yeung. 2017. Spatiotemporal modeling for crowd counting in videos. In *ICCV*. IEEE.

[65] Haipeng Xiong, Hao Lu, Chengxin Liu, Liang Liu, Zhiguo Cao, and Chunhua Shen. 2019. From Open Set to Closed Set: Counting Objects by Spatial Divide-and-Conquer. In *ICCV*. 8362–8371.

[66] Zhaoyi Yan, Yuchen Yuan, Wangmeng Zuo, Xiao Tan, Yezhen Wang, Shilei Wen, and Errui Ding. 2019. Perspective-Guided Convolution Networks for Crowd Counting. In *ICCV*. 952–961.

[67] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. 2017. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *CVPR*. 4133–4141.

[68] Lixian Yuan, Zhilin Qiu, Lingbo Liu, Hefeng Wu, Tianshui Chen, Pei Chen, and Liang Lin. 2020. Crowd counting via scale-communicative aggregation networks. *Neurocomputing* 409 (2020), 420–430.

[69] Sergey Zagoruyko and Nikos Komodakis. 2016. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928* (2016).

[70] B Zhan, Dorothy Monekosso, Paolo Remagnino, Sergio A Velastin, and Liqun Xu. 2008. Crowd analysis: a survey. *Machine Vision Applications* 19, 5 (2008), 345–357.

[71] Anran Zhang, Lei Yue, Jiayi Shen, Fan Zhu, Xiantong Zhen, Xianbin Cao, and Ling Shao. 2019. Attentional Neural Fields for Crowd Counting. In *ICCV*. 5714–5723.

[72] Cong Zhang, Hongsheng Li, Xiaogang Wang, and Xiaokang Yang. 2015. Cross-scene crowd counting via deep convolutional neural networks. In *CVPR*. 833–841.

[73] Feng Zhang, Xiatian Zhu, and Mao Ye. 2019. Fast Human Pose Estimation. In *CVPR*.

[74] Shanghang Zhang, Guanhang Wu, Joao P Costeira, and José MF Moura. 2017. Fcn-rlstm: Deep spatio-temporal neural networks for vehicle counting in city cameras. In *ICCV*. IEEE, 3687–3696.

[75] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. 2018. Deep mutual learning. In *CVPR*. 4320–4328.

[76] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. 2016. Single-image crowd counting via multi-column convolutional neural network. In *CVPR*. 589–597.

[77] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. 2016. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160* (2016).

[78] Michael Zhu and Suyog Gupta. 2017. To prune, or not to prune: exploring the efficacy of pruning for model compression. *arXiv preprint arXiv:1710.01878* (2017).