RECURRENT FINE-GRAINED SELF-ATTENTION NETWORK FOR VIDEO CROWD COUNTING

Jifan Zhang[1], *Zhe Wu*[2], *Xinfeng Zhang*[2,3], *Guoli Song*[2], *Yaowei Wang*[2], *Jie Chen*[1,2]

[1]School of Electronic and Computer Engineering, Peking University, Shenzhen, China
[2]Peng Cheng Laboratory, Shenzhen, China
[3]School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing, China

ABSTRACT

Striking a balance between exploring the spatio-temporal correlation and controlling model complexity is vital for video-based crowd counting methods. In this paper, we propose a Recurrent Fine-Grained Self-Attention Network (RFSNet) to achieve efficient and accurate counting in video scenes via the self-attention mechanism and a recurrent fine-tuning strategy. Specifically, we design a decoder which consists of patch-wise spatial self-attention and temporal self-attention. Compared with vanilla self-attention, it effectively leverages the dependencies in spatial and temporal domain respectively, while significantly reducing computational complexity. Moreover, the RFSNet recurrently feeds the features into the decoder to enhance the spatio-temporal representations. This strategy not only simplifies the model structure and reduces the number of parameters, but also improves the quality of estimated density maps. Our RFSNet achieves state-of-the-art performance on three video crowd counting benchmarks, and outperforms other methods by more than 20% on the challenging FDST dataset.

Index Terms— Crowd counting, temporal modeling, density map regression, self-attention.

1. INTRODUCTION

Crowd counting plays an indispensable role in a lot of computer vision applications, such as auto driving, video surveillance, safety management. Aiming to estimate the accurate number of targets in a single picture or video frames, researchers focus on overcoming challenge issues like scale variation, occlusion and extremely density in the images. Compared with traditional regression-based or detection-based methods, recent counting models based on convolutional neural networks (CNNs) have achieved significant performance improvement. These models introduce multi-branch structure [1–3], dilated convolution [4–6], perspective estimation [7–9], etc., to promote the spatial invariance.

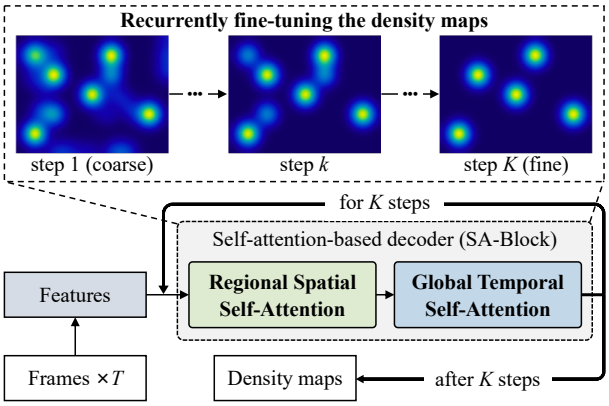

Fig. 1. The self-attention-based decoder is able to capture long-range spatio-temporal dependencies of video frames. And the proposed RFSNet performs a coarse-to-fine process of density maps regression in a recurrent manner.

Nevertheless, most existing methods only consider counting on still images, which results in the neglect of rich temporal information when dealing with video sequences. Therefore, follow-up researchers attempt to capture temporal-wise correlation from long-term and short-term perspective. (1) Long-term methods retain information for long periods of time through global temporal modeling. FCN-rLSTM [10] and ConvLSTM [11] introduce LSTM from sequence learning tasks to model the long-range temporal dependencies. However, they are hard to train and lack of parallelism because of superabundant parameters and the inherent sequential nature of LSTM. Integrated with 3D convolutional layers and channel-wise attention, E3D [12] and STDNet [13] are able to jointly encode global and local spatio-temporal features. But deep 3D CNNs rely heavily on stacked modules, leading to the rapid growth of the network depth and model size. (2) Short-term methods pay attention to the consistency and discrimination between adjacent frames, so as to impose strong smoothness constraints. LSTN [14] use a locality-constrained spatial transformer to estimate the density map

This research was supported in part by the National Key R&D Program of China (Grant No. 2022ZD0118201), the National Natural Science Foundation of China (Grant No. 61972217, 32071459, 62176249, 62006133, 62271465 and 62102207), the Natural Science Foundation of Guangdong Province in China (Grant No. 2019B1515120049), and the Fundamental Research Funds for the Central Universities.

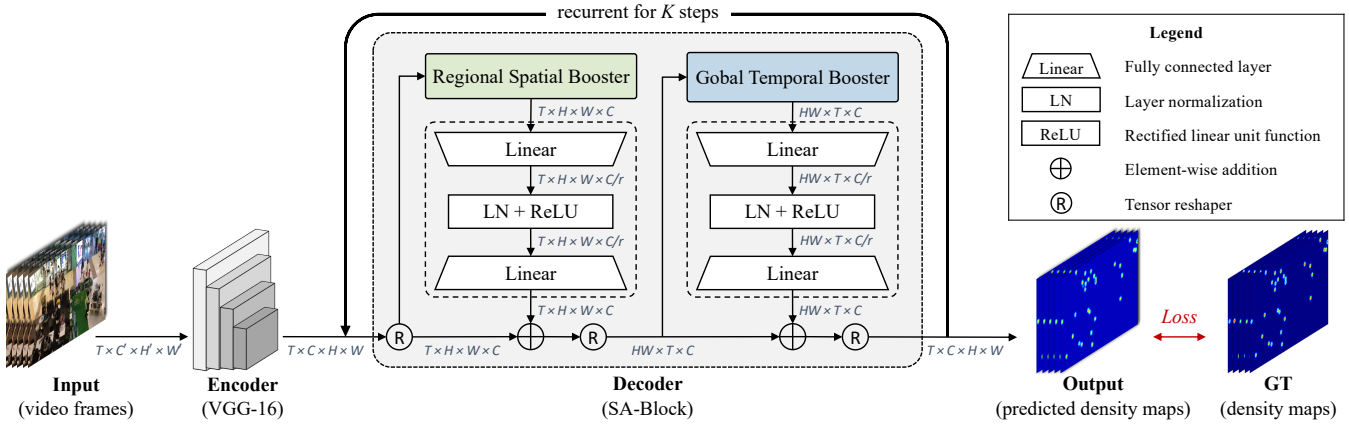

Fig. 2. An overview of the proposed RFSNet. Video frames are fed into an encoder that comprises of the first ten convolutional layers of VGG-16. The resulting features are then recurrently fed into a self-attention-based decoder (SA-Block) to extract spatio-temporal information for K times.

of next frame with that of the current frame. EPF [15] and MOPN [16] introduce optical flow to exploit inter-frame motion clues. Nonetheless, these methods also introduce additional computational overhead for the estimation of flows or spatial transformation mappings, which are lack of annotations for validation and constraint in most cases.

Along with the success of self-attention [17] in sequential learning tasks, there are also some counting methods [18, 19] that takes advantage of its ability in modeling long-distance dependencies. However, global self-attention forms, like non-local [20], roughly represent a mixture of spatial and temporal contexts, leading to the heavy computational complexity and ambiguity for video scene understanding. In addition, crowd images are full of blocks with difference scales but similar patterns, which bring a lot of local redundancy [21]. This further reduces the significance of non-local encoding of primitive self-attention in the spatial domain.

To address these issues, we propose a novel Recurrent Fine-Grained Self-Attention Network (RFSNet) for video crowd counting. As shown in Figure 1, the main component of RFSNet is a self-attention-based decoder, termed SA-Block. It provides the ability to capture long-range spatio-temporal information through self-attention mechanisms specifically acting on spatial and temporal domain, respectively. In order to further reduce the sensitivity of the decoder to scale variations and improve the computational efficiency, we divide the image features into patches and perform spatial self-attention on local regions. Besides, to avoid network deepening and parameter growth, we recurrently feed the output of the SA-Block into itself in a parameter-sharing style [22]. This strategy not only inherits the convergence ability of traditional feed-forward sequential models, but also introduces the recurrent inductive bias of RNNs. Without complicated network structures and additional supervision, our model is able to predict accurate density maps with few parameters.

Our main contributions can be summarized as follows: **(1)** We propose a fine-grained self-attention mechanism to capture spatio-temporal dependencies with high efficiency and low computational complexity. **(2)** We propose a novel Recurrent Fine-Grained Self-Attention Network (RFSNet) which achieves fine-tuning of density maps through a recurrent strategy. **(3)** On three challenging benchmarks, RFSNet demonstrates its effectiveness and achieves new state-of-the-art.

2. METHODS

2.1. Self-Attention and Complexity Analysis

Vanilla Self-Attention. Self-attention, first proposed in [17], calculates a weighted average of feature representations with the weight proportional to a similarity score between representations. Formally, an input sequence of n tokens of dimensions d, $\mathbf{X} \in \mathbb{R}^{n\times d}$ is projected using three matrices $\mathbf{W}^Q \in \mathbb{R}^{d\times d_q}$, $\mathbf{W}^K \in \mathbb{R}^{d\times d_k}$ and $\mathbf{W}^V \in \mathbb{R}^{d\times d_v}$ to extract feature representations $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$. They are also referred to as query, key and value respectively, with $d_k = d_q$. So, self-attention is defined as

$$\mathrm{SA}(\mathbf{X}) = \mathrm{SA}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{softmax}\left(\mathbf{Q}\mathbf{K}^T/\sqrt{d_k}\right)\mathbf{V}, \quad (1)$$

$$\mathbf{Q} = \mathbf{X}\mathbf{W}^Q,\ \mathbf{K} = \mathbf{X}\mathbf{W}^K,\ \mathbf{V} = \mathbf{X}\mathbf{W}^V. \quad (2)$$

To improve the parallelism and learning ability, the queries, keys and values are then projected to h different representation subspaces, named multi-head self-attention,

$$\mathrm{MSA}(\mathbf{X}) = [\mathrm{head}_1,\ldots,\mathrm{head}_h]\mathbf{W}^O, \quad (3)$$

$$\mathrm{head}_i = \mathrm{SA}(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V), \quad (4)$$

where projections are parameter matrices $\mathbf{W}_i^Q \in \mathbb{R}^{d\times d_q}$, $\mathbf{W}_i^K \in \mathbb{R}^{d\times d_k}$, $\mathbf{W}_i^V \in \mathbb{R}^{d\times d_v}$ and $\mathbf{W}_i^O \in \mathbb{R}^{hd_v\times d}$.

Regional Spatial Booster. Global self-attention like non-local [20], in which all spatial locations attend to each other, is too expensive for most image scales due to the quadratic computation cost. Following the intuitive form of local self-attention developed in [23], we design the spatial self-attention based on image patches to improve model efficiency

while focus on the relevance of each location to its neighbor pixels in the same region.

Given an input $\mathbf{X}_S \in \mathbb{R}^{H\times W\times C_{\text{in}}}$, where H is the height, W is the width, and C_{in} is the number of input channels. We divide $\mathbf{X}_S$ into blocks of $p \times p$ and then reshape it to $\mathbf{X}'_S \in \mathbb{R}^{n_H\times n_W\times p^2\times C_{\text{in}}}$, where $n_H = \left\lceil \frac{H}{p} \right\rceil$ and $n_W = \left\lceil \frac{W}{p} \right\rceil$. This operation is denoted as $\overrightarrow{\mathcal{R}}_S$, i.e. $\mathbf{X}'_S = \overrightarrow{\mathcal{R}}_S(\mathbf{X}_S)$. So,

$$\text{RSB}(\mathbf{X}_S) = \overleftarrow{\mathcal{R}}_S\left(\text{MSA}\left(\overrightarrow{\mathcal{R}}_S\left(\mathbf{X}_S\right)\right)\right), \tag{5}$$

where $\text{RSB}(\cdot)$ represents the Regional Spatial Booster, and $\overleftarrow{\mathcal{R}}_S$ is the inverse transformation of $\overrightarrow{\mathcal{R}}_S$. The dimension of the features do not change before and after input.

Global Temporal Booster. Given $\mathbf{X}_T \in \mathbb{R}^{T\times H\times W\times C_{\text{in}}}$, where T is the length of image sequence, $\mathbf{X}'_T \in \mathbb{R}^{HW\times T\times C_{\text{in}}}$ is derived from another reshape operation denoted as $\overrightarrow{\mathcal{R}}_T$, i.e. $\mathbf{X}'_T = \overrightarrow{\mathcal{R}}_T(\mathbf{X}_T)$. So there is

$$\text{GTB}(\mathbf{X}_T) = \overleftarrow{\mathcal{R}}_T\left(\text{MSA}\left(\overrightarrow{\mathcal{R}}_T\left(\mathbf{X}_T\right)\right)\right), \tag{6}$$

where $\text{GTB}(\cdot)$ represents the Global Temporal Booster, and $\overleftarrow{\mathcal{R}}_T$ is the inverse transformation of $\overrightarrow{\mathcal{R}}_T$.

Complexity Analysis. Let $\mathbf{X} \in \mathbb{R}^{T\times H\times W\times C}$ denotes the feature sequences in video scenes. For vanilla self-attention, the computational complexity is $\mathcal{O}\left((THW)^2C\right)$, i.e. $\mathcal{O}\left(THWC \times THW\right)$. However, the computational complexity is reduced to $\mathcal{O}\left(T(HW)^2C + HWT^2C\right)$, i.e. $\mathcal{O}\left(THWC \times (HW + T)\right)$ when we perform self-attention in space domain and time domain separately. Further, after applying $\text{RSB}(\cdot)$ and $\text{GTB}(\cdot)$ mentioned above, the complexity is $\mathcal{O}\left(T \times \frac{HW}{p^2} \times (p \times p)^2 \times C + HWT^2C\right)$, i.e. $\mathcal{O}\left(THWC \times \left(p^2 + T\right)\right)$. Note that the patch size p is usually far less that H and W in experimental settings. So when $T > 1$, $(p^2 + T) \ll (HW + T) < THW$. Hence, our method significantly reduces the computational complexity.

2.2. SA-Block

As shown in Figure 2, the SA-Block can be abstracted into two stages: spatial modeling and temporal modeling.

Spatial Modeling. The first stage can be defined as

$$\mathbf{X}_S = \mathbf{X}_S + \text{Transform}_S\left(\text{RSB}\left(\mathbf{X}_S\right)\right) \tag{7}$$

$$= \mathbf{X}_S + \mathbf{W}_S^2\text{ReLU}\left(\text{LN}\left(\mathbf{W}_S^1\text{RSB}(\mathbf{X}_S)\right)\right), \tag{8}$$

where $\mathbf{X}_S \in \mathbb{R}^{H\times W\times C}$ represents the input spatial features, $\text{RSB}(\cdot)$ denotes the Regional Spatial Booster, LN is the layer normalization and ReLU denotes the rectified linear units. $\mathbf{W}_S^1 \in \mathbb{R}^{C\times\frac{C}{r}}$ and $\mathbf{W}_S^2 \in \mathbb{R}^{\frac{C}{r}\times C}$ are the weights of the two fully connected layers respectively, in which r is the reduction rate for nonlinear dimensionality reduction. The following bottleneck $\text{Transform}_S(\cdot)$ further learns representations of spatial features and fastens the convergence.

Temporal Modeling. Similarly, the second stage is

$$\mathbf{X}_T = \mathbf{X}_T + \text{Transform}_T\left(\text{GTB}\left(\mathbf{X}_T\right)\right) \tag{9}$$

$$= \mathbf{X}_T + \mathbf{W}_T^2\text{ReLU}\left(\text{LN}\left(\mathbf{W}_T^1\text{GTB}(\mathbf{X}_T)\right)\right), \tag{10}$$

Table 1. Comparison of different methods on crowd datasets.

Method	FDST		UCSD		Mall	
	MAE↓	RMSE↓	MAE↓	RMSE↓	MAE↓	RMSE↓
MCNN [1]	3.77	4.88	1.07	1.35	-	-
ConvLSTM [11]	4.48	5.82	1.30	1.79	2.24	8.50
CSRNet [4]	-	-	1.16	1.47	1.70	2.03
LSTN [14]	3.35	4.45	1.07	1.39	2.00	2.50
EPF [15]	2.17	2.62	0.86	1.13	-	-
PHNet [25]	1.65	2.16	0.82	1.05	-	-
MOPN [16]	1.76	2.25	0.97	1.22	1.78	2.25
STDNet [13]	-	-	0.76	1.01	1.47	**1.88**
RFSNet (ours)	**1.25**	**1.60**	**0.72**	**0.91**	**1.46**	1.90

where $\mathbf{X}_T \in \mathbb{R}^{T\times H\times W\times C}$ denotes the input temporal features, $\text{GTB}(\cdot)$ denotes the Global Temporal Booster, $\mathbf{W}_T^1 \in \mathbb{R}^{C\times\frac{C}{r}}$ and $\mathbf{W}_T^2 \in \mathbb{R}^{\frac{C}{r}\times C}$ are the weights of the two fully connected layers respectively.

2.3. Recurrent Fine-Grained Self-Attention Network

Following previous works [4, 13, 14], we choose the first ten convolutional layers of VGG-16 [24] as the encoder because of its strong transfer learning ability. Then, the decoder (SA-Block) takes the features to extract high-level spatio-temporal information while retaining the size of feature maps. Inspired by [22], we recurrently feed the features into the same SA-Block, instead of stacking several SA-Blocks to construct a more bloated decoder, as shown in Figure 2.

After cycles of K times, a regression head consisting of three convolution layers performs feature dimension reduction and density maps generation. In the end, the estimated density maps, after resampling to the same size as the input, and the ground truth are fed into the loss function with a gradient-descent-based optimizer to train the parameters.

2.4. Loss Function

Following previous works [1, 4, 11, 14], we adopt the Euclidean distance to measure the pixel-wise difference between estimated density maps and their corresponding ground truth. Thus, the regression loss is defined as

$$\mathcal{L}(\Theta) = \frac{1}{2N}\sum_{i=1}^{N}\left\|\mathbf{D}^{\text{EST}}\left(\mathbf{I}_i;\Theta\right) - \mathbf{D}^{\text{GT}}(\mathbf{I}_i)\right\|_2^2, \tag{11}$$

where N is the size of training set and $\mathbf{I}_i$ represents the i-th input image. $\mathbf{D}^{\text{EST}}\left(\mathbf{I}_i;\Theta\right)$ is the output of RFSNet with parameters Θ, while $\mathbf{D}^{\text{GT}}(\mathbf{I}_i)$ is its corresponding ground truth.

3. EXPERIMENTS

3.1. Experimental Settings and Datasets

Implementation Details. We use the pre-trained weights of the encoder to accelerate the training process instead of training from scratch. Adam [26] is utilized as the optimizer to minimize $\mathcal{L}$, while the learning rate is set to 1×10^{-4} initially. We fix the number of recursion $K = 4$ and the batch size is always set to 1. We set the patch size $p = 5$, and the time step $T = 5$. Additionally, random resized cropping and horizontal flipping are used to perform data augmentation.

Datasets. We conduct exhaustive experiments on three video-based crowd datasets. FDST [14] contains 15000 images

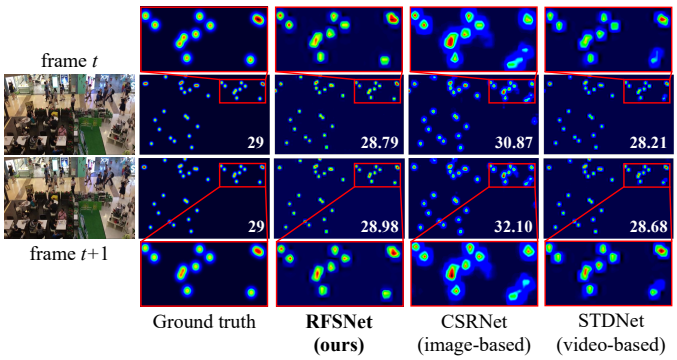

Fig. 3. Qualitative analysis on consecutive frames of FDST dataset. Each column represents the input image, ground truth, and predicted density maps of the proposed RFSNet, CSRNet [4] and STDNet [13] respectively. Counting results are marked in the lower right corner of each density map. Regions with greatest differences are marked by red boxes.

taken from 13 different scenarios, 9000 of which for training and the rest for test. UCSD [27] is composed of 10 video clips with a total of 2,000 frames captured by a stationary camera in a fixed scene. MALL [28] consists of 2,000 frames captured in a shopping mall with one surveillance camera.

Evaluation Matrics. Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) are adopted as the evaluation metrics, same as previous works [1–6, 10–13].

3.2. Experimental Results

Quantitative Results. The comparison of different models on the three crowd counting datasets are shown in Table 1. In terms of counting accuracy, the proposed RFSNet is superior to all previous methods. Especially compared with MOPH [16], our method has more than 20% improvement in both MAE and MSE metrics on all three datasets.

Qualitative Analysis. In Figure 3, we provide a qualitative comparison between RFSNet and other methods. It is evident that our model has the ability to estimate an accurate density map while precisely calculating the count. Especially for the same regions between adjacent frames, the prediction results of RFSNet are more stable and consistent. Compared with others, RFSNet has more accurate estimation results for both the overall count and the details of the density map.

3.3. Ablation Study

Effectiveness of SA-Block. As shown in Table 2, the SA-Block without self-attention mechanism (Exp. II) has significantly improved performance over the baseline (Exp. I) on FDST dataset. The results are further improved when the RSB and GTB are plugged in (Exp. III, IV and V).

Effectiveness of the recurrent strategy. As shown in Table 3, when we sequentially stack the SA-Block (Exp. VI, VII and VIII), the number of model parameters grows significantly. However, in the recurrent cases (Exp. IX, X and XI),

Table 2. Ablation study of components of the SA-Block.

Exp.	Setting			Performance	
	SA-Block	RSB	GTB	MAE↓	RMSE↓
I				1.57	2.13
II	√			1.42	1.86
III	√	√		1.39	1.85
IV	√		√	1.35	1.83
V	√	√	√	**1.25**	**1.60**

Table 3. Ablation study of SA-Block arrangement policies.

Exp.	**Setting**			**Performance**	
	SA-Block	K	#Params (M)	MAE↓	RMSE↓
VI	Sequential	2	9.879	1.34	1.83
VII		4	11.459	1.31	1.75
VIII		6	13.039	1.36	1.77
IX	Recurrent	2	**9.089**	1.29	1.70
X		4	**9.089**	**1.25**	**1.60**
XI		6	**9.089**	1.30	1.65

Table 4. Comparison of efficiency of different methods.

Method	#Params (M)	Training speed (s/epoch↓)	Inference speed (fps↑)	Acc. (MAE↓)
ConvLSTM [11]	40.610	530	13	1.30
STDNet [13]	18.146	50	**34**	0.76
RFSNet + NL	**8.694**	438	4	1.21
RFSNet + SSA + GTB	9.089	115	17	0.87
RFSNet + RSB + GTB	9.089	**46**	23	**0.72**

the change of K has no effect on the model size. It is obvious that when K increases to a certain extent, the performance of the model on FDST dataset does not increase proportionable, but the calculation efficiency is greatly reduced. As a trade-off between accuracy and efficiency, K is set to 4.

Discussion of model size and efficiency. Owing to our recurrent strategy, the proposed RFSNet is lightweight in model size. In Table 4, we compare it with other video-based methods on UCSD dataset. The number of parameters of RFSNet is 9.089 million only, saving nearly 50% compared to STDNet [13] and more than 75% compared to ConvLSTM [11]. Compared to the global self-attention mechanisms acting directly on the spatio-temporal domain (NL) or the spatial domain (SSA), the combination of RSB and GTB achieves remarkable leap of efficiency and accuracy.

4. CONCLUSION

In this paper, we proposed a novel Recurrent Fine-Grained Self-Attention network (RFSNet) to solve the task of video-based crowd counting. Through a decoder consisting of patch-wise spatial self-attention and temporal self-attention, as well as a recurrent strategy, RFSNet effectively leverages the spatio-temporal correlation between video frames and generates fine-tuned density maps. The experiments conducted on three video counting datasets demonstrate that our method achieves state-of-the-art performance and maintains computational efficiency.

5. REFERENCES

[1] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-Image Crowd Counting via Multi-Column Convolutional Neural Network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2016, pp. 589–597.

[2] X. Cao, Z. Wang, Y. Zhao, and F. Su, "Scale Aggregation Network for Accurate and Efficient Crowd Counting," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 734–750.

[3] Y. Yang, G. Li, D. Du, Q. Huang, and N. Sebe, "Embedding Perspective Analysis Into Multi-Column Convolutional Neural Network for Crowd Counting," *IEEE Transactions on Image Processing*, vol. 30, pp. 1395–1407, 2021.

[4] Y. Li, X. Zhang, and D. Chen, "CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 1091–1100.

[5] D. Guo, K. Li, Z.-J. Zha, and M. Wang, "DADNet: Dilated-Attention-Deformable ConvNet for Crowd Counting," in *Proceedings of the ACM International Conference on Multimedia*, Oct. 2019, MM '19, pp. 1823–1832.

[6] S. Bai, Z. He, Y. Qiao, H. Hu, W. Wu, and J. Yan, "Adaptive Dilated Network With Self-Correction Supervision for Counting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4594–4603.

[7] M. Shi, Z. Yang, C. Xu, and Q. Chen, "Revisiting Perspective Information for Efficient Crowd Counting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2019, pp. 7271–7280.

[8] Y. Yang, G. Li, Z. Wu, L. Su, Q. Huang, and N. Sebe, "Reverse Perspective Network for Perspective-Aware Object Counting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2020, pp. 4373–4382.

[9] Z. Yan, R. Zhang, H. Zhang, Q. Zhang, and W. Zuo, "Crowd Counting via Perspective-Guided Fractional-Dilation Convolution," *IEEE Transactions on Multimedia*, pp. 2633–2647, 2021.

[10] S. Zhang, G. Wu, J. P. Costeira, and J. M. F. Moura, "FCN-rLSTM: Deep Spatio-Temporal Neural Networks for Vehicle Counting in City Cameras," in *Proceedings of the International Conference on Computer Vision*, Oct. 2017, pp. 3687–3696.

[11] F. Xiong, X. Shi, and D.-Y. Yeung, "Spatiotemporal Modeling for Crowd Counting in Videos," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5151–5159.

[12] Z. Zou, H. Shao, X. Qu, W. Wei, and P. Zhou, "Enhanced 3D convolutional networks for crowd counting," in *Proceedings of the British Machine Vision Conference*, Sept. 2019, pp. 250–260.

[13] Y.-J. Ma, H.-H. Shuai, and W.-H. Cheng, "Spatiotemporal Dilated Convolution With Uncertain Matching for Video-Based Crowd Estimation," *IEEE Transactions on Multimedia*, vol. 24, pp. 261–273, 2022.

[14] Y. Fang, B. Zhan, W. Cai, S. Gao, and B. Hu, "Locality-Constrained Spatial Transformer Network for Video Crowd Counting," in *Proceedings of the International Conference on Multimedia and Expo*, July 2019, pp. 814–819.

[15] W. Liu, M. Salzmann, and P. Fua, "Estimating People Flows to Better Count Them in Crowded Scenes," in *Proceedings of the European Conference on Computer Vision*, 2020, pp. 723–740.

[16] M. A. Hossain, K. Cannons, D. Jang, F. Cuzzolin, and Z. Xu, "Video-Based Crowd Counting Using a Multi-Scale Optical Flow Pyramid Network," in *Proceedings of the Asian Conference on Computer Vision*, 2020, pp. 3–20.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, 2017, vol. 30, pp. 5998–6008.

[18] Q. Wu, C. Zhang, X. Kong, M. Zhao, and Y. Chen, "Triple Attention For Robust Video Crowd Counting," in *Proceedings of the International Conference on Image Processing*, 2020, pp. 1966–1970.

[19] H. Bai and S.-H. G. Chan, "Motion-guided Non-local Spatial-Temporal Network for Video Crowd Counting," *CoRR*, vol. abs/2104.13946, 2021.

[20] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-Local Neural Networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7794–7803.

[21] V. Lempitsky and A. Zisserman, "Learning To Count Objects in Images," in *Proceedings of the Advances in Neural Information Processing Systems*, 2010, pp. 1324–1332.

[22] M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, and L. Kaiser, "Universal Transformers," in *International Conference on Learning Representations*, 2019.

[23] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "Stand-Alone Self-Attention in Vision Models," in *Advances in Neural Information Processing Systems*, 2019, vol. 32, pp. 68–80.

[24] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *Proceedings of the International Conference on Learning Representations*, May 2015.

[25] S. Meng, J. Li, W. Guo, L. Ye, and J. Jiang, "PHNet: Parasite-Host Network for Video Crowd Counting," in *Proceedings of the International Conference on Pattern Recognition*, 2020, pp. 1956–1963.

[26] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proceedings of the International Conference on Learning Representations*, May 2015.

[27] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2008, pp. 1–7.

[28] K. Chen, C. C. Loy, S. Gong, and T. Xiang, "Feature Mining for Localised Crowd Counting," in *Proceedings of the British Machine Vision Conference*, 2012, pp. 21.1–21.11.