Enhanced 3D convolutional networks for crowd counting

Zhikang Zou*[1]
zhikangzou001@gmail.com
Huiliang Shao*[1]
u201615039@hust.edu.cn
Xiaoye Qu[1]
xiaoye@hust.edu.cn
Wei Wei[2]
weiw@hust.edu.cn
Pan Zhou†[1]
panzhou@hust.edu.cn

[1] School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China
[2] School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China

Abstract

Recently, convolutional neural networks (CNNs) are the leading defacto method for crowd counting. However, when dealing with video datasets, CNN-based methods still process each video frame independently, thus ignoring the powerful temporal information between consecutive frames. In this work, we propose a novel architecture termed as "temporal channel-aware" (TCA) block, which achieves the capability of exploiting the temporal interdependencies among video sequences. Specifically, we incorporate 3D convolution kernels to encode local spatio-temporal features. Furthermore, the global contextual information is encoded into modulation weights which adaptively recalibrate channel-aware feature responses. With the local and global context combined, the proposed block enhances the discriminative ability of the feature representations and contributes to more precise results in diverse scenes. By stacking TCA blocks together, we obtain the deep trainable architecture called enhanced 3D convolutional networks (E3D). The experiments on three benchmark datasets show that the proposed method delivers state-of-the-art performance. To verify the generality, an extended experiment is conducted on a vehicle dataset TRANCOS and our approach beats previous methods by large margins.

1 Introduction

Crowd counting algorithms aim to produce an accurate estimation of the true number of individuals in a still image or a video. It has drawn much attention due to the important geo-political and civic applications in video surveillance, traffic control, and abnormally detection. Moreover, some crowd counting methods with great generality can be extended to

* Equal contribution
† Corresponding author

other applications, such as automobile counting at traffic jams, cell or bacteria counting from microscope images and animal estimation in wild scenes. However, it's still a challenging vision task to obtain accurate individual number because of severe occlusion, diverse distribution and perspective distortion.

Recently, researchers have leveraged Convolutional Neural Networks (CNNs) for an accurate crowd density map generation [2, 16, 33, 37]. Some works focus on explicitly incorporating multi-scale information based on multi-column architectures [1, 36]. They use different filter sizes for different columns which are adaptive to the scale variation in crowd size. Instead, FCN-7c [13] feeds an image pyramid of the input image into a single column network in order to alleviate heavy computational overhead. Though these methods have made significant progress, they are restricted by capturing the informative representations with local receptive fields, which isolates the pixels from the global scene context. In CP-CNN [25], Contextual Pyramid CNNs are proposed to explicitly incorporate global and local contextual information of crowd images which are fused with high-dimensional feature maps to generate accurate density maps. However, they need to train two additional networks to evaluate the context of crowds, which suffers from high computation complexity. Moreover, existing CNN-based methods are faced with inherent algorithmic weaknesses: when dealing with video sequences, they still regard the data as single still images, thus ignoring the temporal information stored in neighbouring frames. To exploit the strong correlation in video data, Xiong *et al*. [32] propose a variant of LSTM in order to process the sequence of images into density maps. However, that LSTM is difficult to train hinders the wide application of the proposed method.

To practically resolve these problems, we propose a novel temporal channel-aware (TCA) block to utilize the correspondence among video sequences and capture global feature statistics simultaneously. Motivated by the achievement of 3D CNN in action recognition [26], we employ 3D convolutions in the proposed block to encode spatio-temporal features for videos, especially improving the representation ability in temporal dimension. Besides, global contextual information is transformed into modulation weights which adaptively highlight the useful features by rescaling each channel-aware feature response. We also leverage short skip connections to ease the training of the model. Therefore, we could stack several TCA blocks together to form very deep network, named as enhanced 3D convolutional networks (E3D). Extensive experiments on three benchmark datasets (inculding WorldExpo'10 [33], UCSD [5] and MALL [6]) show that the proposed E3D yields significant improvement over recent state-of-the-art methods. Furthermore, we evaluate E3D on a vehicle dataset TRANCOS [10] to demonstrate the generality for counting other objects.

To summarize, we make the following contributions:

- To the best of our knowledge, it's the first attempt to adopt 3D convolution for crowd counting. By introducing 3D kernels, the model is capable of capturing the temporal and spatial information simultaneously, thereby boosting the performance on video datasets.

- We design a novel temporal channel-aware (TCA) block to incorporate both local and global spatio-temporal information. By applying the proposed block into the enhanced 3D convolutional networks (E3D), the network achieves tremendous improvement in strengthening the discriminative ability of feature representations.

- We conduct a throughout study on the number of frames sent to network, the number of TCA blocks and the components of the whole architecture.

2 Related Work

Plenty of algorithms have been proposed for crowd counting to solve related real world problems [17, 20, 22, 24]. Most of the early researchers focus on detection-based framework using a moving-window-like detector to estimate the number of individuals. These methods require well-trained classifiers to extract low-level features from a full body such as Haar Wavelets [28] and HOG [7]. However, for crowded scenarios, objects are highly occluded and difficult to detect. To tackle this problem, researchers have attempted to detect particular body parts instead of the whole body to estimate the count [14, 31]. For instance, Li *et al*. [14] incorporate a foreground segmentation algorithm and a HOG-based head-shoulder detection algorithm to detect heads, thus implementing crowd scenes analysis.

Although the part-based detection methods alleviate the problem of occlusion, they perform poorly on extremely congested scenes and high background clutter scenes. Researchers make an effort to deploy regression-based approaches which learn a mapping between extracted features from cropped images patches and their count or density [5, 6]. Moreover, some methods [4, 9] leverage spatial or depth information and take approach of segmentation methods to filter the background region, thereby regressing count numbers only on foreground segments. These methods are sensitive to different crowd density and have addressed the problem of severe occlusion in dense crowds.

Recently, CNN-based approaches have become ubiquitous owing to its success in a large number of computer vision tasks [18, 27, 35]. Many researchers have shifted their attention towards CNN-based methods, which have achieved significant improvements over previous methods of crowd counting. Zhang *et al*. [36] propose a multi-column based architecture to tackle the large scale variations in crowd scenes. Similarly, Onoro *et al*. [19] develope a scale-aware counting model called Hydra CNN for object estimation. Sam *et al*. [1] use a switch layer to select the most optimal regressor for the particular input patches. In order to pursue the quality of the density maps, Sindagi *et al*. [25] propose contextual pyramid networks to generate high-quality density maps by explicitly combining global and local contextual information at the expense of complicated structures. However, multi-column structures are much more difficult to train because each subnet may have different loss surfaces. Therefore, SCNet [30] makes a balance between pixel-wise estimation and computational costs by designing a single-column network. Further, Li *et al*. [15] incorporate dilated convolutions to aggregate multi-scale contextual information.

Despite the promising results, all the methods mentioned above neglect the otherwise powerful temporal information when dealing with video data. To deal with this problem, some methods [32, 34] attempt to utilize variants of LSTM to access long-range temporal dependencies. However, the complex LSTM architecture indicates the requirement of heavy computational costs and difficulty of training relevant parameters. Instead, we introduce 3D convolutions to exploit temporal information among videos in this paper. A novel architecture termed as "temporal channel-aware" (TCA) block is designed to capture temporal interdependencies and encode global contextual information simultaneously.

3 Our approach

Existing CNN-based crowd models mostly fail to take the temporal information into account for those images captured from video datasets. To overcome this issue, our solution is to adopt enhanced 3D convolutional network stacked by temporal channel-aware block to

capture the strong temporal correlation.

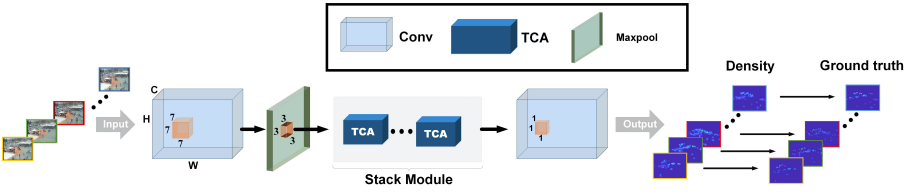

Figure 1: Network architecture of our enhanced 3D convolutional network.

3.1 Backbone architecture

The architecture of the proposed E3D network is shown in Fig. 1. Given multiple successive frames stacked as inputs, this sequence is first processed by a 3D convolution with a kernel size of 7x7x7 and stride 1x2x2. With the help of the sliding convolution operation among the time dimension, the temporal correlation between neighbouring input frames is captured. Then a max-pooling with pooling size of 3x3x3 and stride 1x1x1 is applied to downsample the extracted feature maps. To further explore the spatio-temporal information, we stack eight TCA blocks which can analyze the features from both local and global perspectives. The number of TCA blocks is chosen according to the performance on all testing datasets. There are two different types of TCA blocks used in the stream with slight nuance as one type changes the stride of the first convolution to 1x2x2 in order to downsample features in the spatial dimension while the others stay 1x1x1. We stack these two TCA blocks alternately and make sure that the output of the stack module is 1/16 of the input size. The detailed description of the TCA block will be introduced in the next section. After this stage, we use a 3D convolution with a kernel size of 1x1x1 to match the channel of output feature maps with the ground-truth density maps. Meanwhile, we scale the ground-truth density maps using bilinear interpolation with the factor of 1/16 to match the size of the final density maps. The proposed E3D is a fully convolutional network to accept inputs of arbitrary sizes and can be optimized via an end-to-end training scheme.

3.2 TCA Block

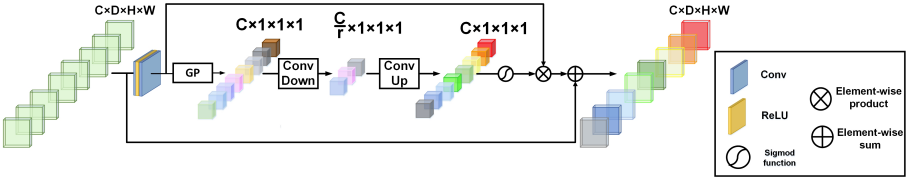

Figure 2: Temporal channel-aware block

The critical component of our architecture is the TCA block, as is depicted in Fig. 2. This block can be divided into two branches, namely the mainstream and the shortcut branch. The mainstream branch deals with the feature maps and reconstructs the channel-wise feature response, while the shortcut branch here is to leverage the effectiveness of the residual learning

for effective training whose spirit is similar to ResNet [11]. Specifically, in the mainstream branch, the feature maps X input to the block first pass two 3D convolutional layers. In this transformation, the number of channels remains unchanged. We use c and c' to represent the number of input and output channels, respectively. For simplicity, bias terms are omitted and each 3D convolutional transformation can be formulated as:

$$o_{c'} = k_{c'} * X = \sum_{i=1}^{c} k_{c'}^{i} * x^{i}, \tag{1}$$

where * denotes convolution, $k_{c'} = [k_{c'}^{1}, k_{c'}^{2}, \cdots k_{c'}^{c}]$ is the set of learned filter kernels and $k_{c'}^{i}$ is a 3D spatial kernel for a single input channel, $X = [x^1, x^2, \cdots x^c]$ are input feature maps. Because c' is equal to c, thus we still use c for following notations. After the process of two convolutions and a relu, the output features O can be denoted as $O = [o_1, o_2, \cdots o_c]$. Global contextual information is expected to be fused into both spatial and temporal dimensions because of the specific informative features in each channel, which indicates that it's inapposite to treat all channels with equality. Therefore, we send the output features to a channel descriptor by aggregating feature maps across their spatio-temporal dimensions. The channel descriptor is produced by global average pooling to generate channel-wise weights so that our network can selectively increase its sensitivity to useful informative features which can be effectively exploited by subsequent transformations. Formally, channel-wise means v_c are generated by shrinking O through spatiotemporal dimensions $D \times H \times W$:

$$v_c = \frac{1}{D \times H \times W} \sum_{l=1}^{D} \sum_{m=1}^{H} \sum_{n=1}^{W} o_c(l,m,n). \tag{2}$$

To further exploit the channel dependencies, we use a dimensionality-reduction convolution layer followed by a dimensionality-increasing convolution layer to automatically learn the subtle interaction relationships between channels. We then utilize a sigmoid activation to normalize the weights of multiple channels. This procedure can be defined as:

$$u = \sigma(W_2 \delta(W_1 v)), \quad v = [v_1, v_2, \cdots, v_c], \tag{3}$$

where $u = [u_1, u_2, \cdots, u_c]$ is the normalized weight, δ refers to the ReLU function, σ refers to the sigmoid function, $W_1 \in R^{\tilde{C} \times C}$, $W_2 \in R^{C \times \tilde{C}}$ are the convolutions, $\tilde{C} = \frac{C}{r}$ and r is the reduction ratio. In this paper, r is set to 4. With the normalized channel weights, the channel information in the TCA block is adaptively rescaled. The output of the mainstream branch can be achieved by channel-wise multiplication between the feature map $o_c \in R^{D \times H \times W}$ and the normalized channel weight u_c. The process of producing output can be formulated as:

$$\tilde{O} = [\tilde{o}_1, \tilde{o}_2, \cdots, \tilde{o}_c], \quad \tilde{o}_c = o_c \cdot u_c. \tag{4}$$

Finally, by integrating information from both branches, we can get the final output of TCA block as:

$$\tilde{X} = X + \tilde{O}. \tag{5}$$

3.3 TCA-2D: a degenerate variant of TCA block

To get a further understanding of the effectiveness of exploiting temporal information, we propose a degenerate variant of TCA block in a 2D version called TCA-2D, which is stacked

to form enhanced 2D convolutional networks termed as E2D. Except for the downsampling layer, we replace all the 3D kernels with the corresponding 2D ones. With regard to the downsampling layers with stride 1x2x2, we use 2x2 to replace the original stride.

In the experiments to be reported in the next section, whenever the dataset consists of images not captured from the same video sequences, the E3D will not come into effect but only E2D will be employed.

3.4 Ground Truth Generation

Following the method of generating density maps in [33], we generate the ground truth by blurring each head annotations via a Gaussian kernel which is normalized to sum to one. Therefore, the total sum of the density map equals to the actual crowd counts. The ground truth G is given as follow:

$$G(x) = \sum_{i=1}^{N} \delta(x - x_i) * G_\sigma(x), \tag{6}$$

where N is the total number of the individuals and $G_\sigma(x)$ represents 2D Gaussian kernels. Considering the negative effect of perspective distortion to some extent, we employ the geometry-adaptive kernels [36] to process the datasets lack of geometry information. The geometry-adaptive kernels are defined as:

$$F(x) = \sum_{i=1}^{N} \delta(x - x_i) \times G_{\sigma_i}(x), \qquad \sigma_i = \beta \bar{d}^i. \tag{7}$$

For each head x_i, we use $\bar{d}^i$ to indicate the average distance of k nearest neighbours. $\delta(x - x_i)$ is convolved with a Gaussian kernel with standard deviation parameter σ_i where x is the position of pixel in each image. In the experiment, the ratio β is set to 0.3 and k is 3.

4 Experiment

We demonstrate the effectiveness of the proposed E3D model on three popular video datasets as well as the vehicle dataset TRANCOS. Some statistics of these datasets and the corresponding kernels we use are summarized in Table 1. Besides, ablation studies are conducted on the UCSD dataset to analyze the impact of the number of video frames sent to the network, the effect of the number of the stacked TCA blocks and the capability of each component in our network. Qualitative results are visualized in Fig. 3.

Dataset	Resolution	Color	Num	FPS	Max	Min	Average	Total	Generating method
UCSD [8]	238 × 158	Grey	2000	10	46	11	24.9	49885	Fixed kernel: σ=4
Mall [8]	640 × 480	RGB	2000	< 2	53	11	31.2	62315	Geometry-adaptive kernels
WorldExpo'10 [33]	720 × 572	RGB	3980	50	253	1	50.2	199923	Fixed kernel: σ=3
TRANCOS [10]	640 × 480	RGB	1244	-	-	-	37.6	46796	Fixed kernel: σ=4

Table 1: Statistics of the four datasets

4.1 Evaluation metrics

The widely used *mean absolute error* (MAE) and the *mean squared error* (MSE) are adopted to evaluate the performance of different methods. The MAE and MSE are defined as follows:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|q_i - \hat{q}_i|, \qquad \text{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(q_i - \hat{q}_i)^2}. \tag{8}$$

Here, N represents the total number of frames in the testing datasets, q_i and $\hat{q}_i$ are the ground truth and the estimated count, respectively. $\hat{q}_i$ is calculated by summing up the estimated density map over the entire image.

4.2 Results

UCSD. The UCSD crowd counting dataset [5] consists of 2000 video frames of pedestrians on a walkway of the UCSD campus captured by a stationary camera. The video was recorded at 10fps with dimension 238 × 158. A region of interest(ROI) is provided for the scene in the dataset so that all frames and corresponding ground truth are masked with ROI. In the final output feature map, the intensities of pixels out of ROI is also set to zero, thereby constraining the error to the ROI areas to backpropagate during training. Following the setting in [5], we use frame 601-1400 as the training data and the remaining 1200 frames as test data. We adopt a fixed spread Gaussian to generate ground truth density maps for training the network as the crowd is relatively sparse. Considering the fact that the resolution of each frame is small and fixed, each image is resized to two times the original size before it is sent to the network. We send 16 frames to the network at a time. The results of the different methods are shown in Table 2. Compared with other state-of-the-art approaches, our method achieves the best result, which can be regarded as a verification that temporal information can boost the performance for this dataset.

Method	MAE	MSE
Cross-Scene [33]	1.60	3.31
MCNN [36]	1.07	1.35
Bidirectional ConvLSTM [32]	1.13	1.43
Switching-CNN [1]	1.62	2.10
CSRNet [15]	1.16	1.47
SANet [3]	1.02	1.29
E3D (ours)	**0.93**	**1.17**

Table 2: Performance on UCSD dataset

Method	MAE	MSE
Ridge regression [6]	3.59	19.1
Count forest [20]	2.50	10.0
Weighted VLAD [23]	2.41	9.12
Exemplary Density [29]	1.82	2.74
MCNN [36]	2.24	8.5
Bidirectional ConvLSTM [32]	2.10	7.6
E3D (ours)	**1.64**	**2.13**

Table 3: Performance on Mall dataset

Mall. The mall dataset [6] was provided by Chen *et al.* for crowd counting. It was collected from a public accessible webcam in a shopping mall. The video contains 2000 annotated frames of over 60000 pedestrians with their head positions labeled. The ROI is also provided in the dataset. We use the first 800 frames for training and the remaining 1200 frames for testing. With more challenging lighting conditions and glass surface reflection, it's difficult to find the underlying relationship between the head size and density map. Thus geometry-adaptive kernels are applied to generate the density map. Also, 16 frames are sent to the network simultaneously. We perform a comparison against previous methods and our method achieves state-of-the-art performance with respect to both MAE and MSE. The results are shown in Table 3, which also verifies the effectiveness of the powerful temporal information between recurrent frames.

WorldExpo'10. The WorldExpo'10 dataset [33] is made up of 3980 annotated frames from 1132 video sequences captured by 108 different surveillance cameras. This dataset is split

into a training set of 3380 frames collected by 103 cameras and a testing set of 600 frames from 5 different scenes. The region of interest (ROI) are provided for these five test scenes. Each frame and its dot maps are masked with ROI during processing. We still use 16 frames as inputs and the MAE metric for evaluation. As shown in Table 4, the proposed E3D achieves the best accuracy in 4 out of 5 scenes and delivers the lowest average MAE compared with previous methods.

TRANCOS. In addition to counting pedestrians, an extended experiment is conducted on the vehicle dataset TRANCOS [10] to present the generality of our model. It consists of 1244 images of different congested traffic scenes with a total of 46796 vehicles annotated. Different from crowd counting datasets, TRANCOS contains multiple scenes from different video sequences, which indicates intercepted frames are not consecutive. Since there is no temporal correlation between them, we can not make use of the advantage of 3D convolutions. Therefore, we degenerate the 3D model into a 2D version named E2D, which is made up of TCA-2D blocks. Strictly following the setting in [10], we adopt the Grid Average Mean absolute Error (GAME) metric, which is:

$$\mathrm{GAME}(L) = \frac{1}{N}\sum_{n=1}^{N}\Big(\sum_{l=1}^{4^L}\Big|C_{I_n}^{l} - C_{I_n}^{l_{GT}}\Big|\Big), \tag{9}$$

where N is the number of test images, $C_{I_n}^{l}$ is the estimated count of image n within region l and $C_{I_n}^{l_{GT}}$ is the corresponding ground truth result. The GAME metric aims at subdividing the image into 4^L non-overlapping region and evaluating the accuracy of the estimated position. When L=0, the GAME is equivalent to MAE. We compare our approach with five previous methods in Table 5 and achieve a significant improvement in four different GAME metrics. This illustrates that our model can still obtain robust results in the absence of temporal information.

Method	S1	S2	S3	S4	S5	Avg
Cross-Scene [33]	9.8	14.1	14.3	22.2	3.7	12.9
MCNN [34]	3.4	20.6	12.9	13.0	8.1	11.6
Switching-CNN [1]	4.4	15.7	10.0	11.0	5.9	9.6
CP-CNN [25]	2.9	14.7	10.5	10.4	5.8	8.86
SCNet [30]	**1.8**	9.6	14.2	13.3	3.2	8.4
E3D (ours)	2.8	12.5	12.9	**10.2**	**3.2**	**8.32**

Table 4: Performance on WorldEXpo'10

Method	GAME0	GAME1	GAME2	GAME3
Hydra 3s [19]	10.99	13.75	16.69	19.32
FCN-HA [24]	4.21	-	-	-
AMDCN [8]	9.77	13.16	15.00	15.87
FCNN-skip[23]	4.61	8.39	11.08	16.10
CSRNet [15]	3.56	5.49	8.57	15.04
E2D (ours)	**2.88**	**4.81**	**7.77**	**12.47**

Table 5: Performance on TRANCOS

Method	MAE	MSE	Frame length	MAE	MSE	TCA numbers	MAE	MSE
E2D (w/o gc)	1.10	1.40	4	1.26	2.79	4	1.18	1.48
E2D	1.00	1.31	8	1.27	2.32	6	1.12	1.45
E3D (w/o gc)	1.00	1.33	12	1.08	1.40	8	**0.93**	**1.13**
E3D	**0.93**	**1.13**	16	**0.93**	**1.13**	10	1.02	1.35

Table 6: The ablations on UCSD about the capability of each component in our network, the impact of the number of video frames sent to the network and the effect of the number of the stacked TCA blocks.

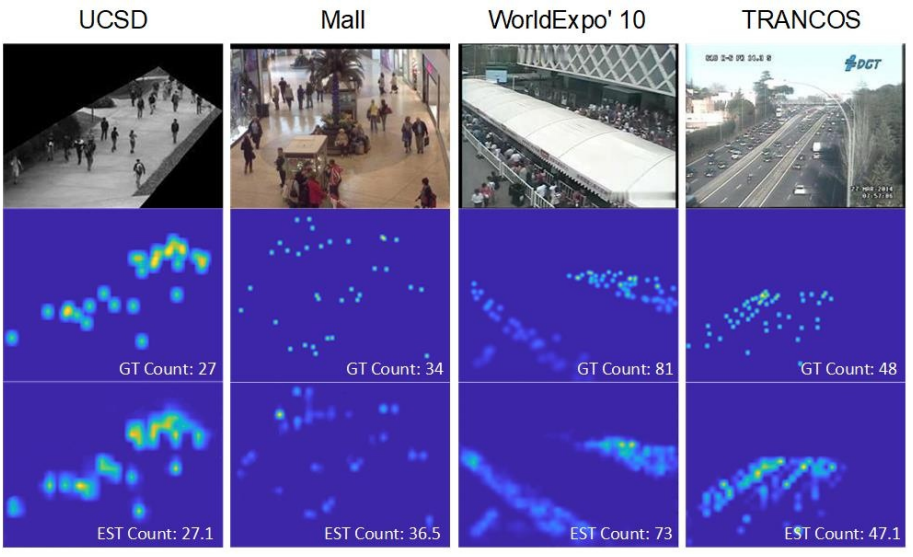

Figure 3: Visualization of estimated density maps on four benchmark datasets by the proposed enhanced 3D convolutional networks.

4.3 Ablation study

To have more insights into our proposed method, we conduct ablation studies on UCSD datasets for its representative temporal information.

1) Component analysis: Our first study is to investigate the capability of each component in the proposed E3D and the results are listed in Table 6 (left). The first method E2D (w/o gc) means that we remove the mainstream branch (the global context branch) in each TCA-2D block of the E2D, the same goes for E3D (w/o gc) based on E3D. From the table, we could see that incorporating global context can reduce the MAE from 1.10 (E2D (w/o gc)) to 1.00 (E2D) or from 1.00 (E3D (w/o gc)) to 0.93 (E3D), which demonstrates its effectiveness in improving the performance of the proposed model. Besides, the performance discrepancy between E2D (w/o gc) and E3D (w/o gc) or E2D and E3D illustrates that it is useful to exploit the temporal information in video sequences, which supports our justification for the use of the 3D convolutions.

2) Frame length: Given the benefits of temporal convolutions above, it is interesting to study the impact of the number of the frames sent to the network on the final performance. As shown in Table 6 (middle), we gradually increase the number of frames at intervals of 4. It is obvious that the results are comparative in the case of frames 4 and 8 (MAE 1.26 vs 1.27). However, the network achieves significant performance improvement when the number of frames is 12 or 16. It is because the UCSD dataset is recorded at 10fps. There is little temporal information to make use of when frame length is less than 10. This may suggest that our model benefits from the increase of the number of frames sent to the network. Considering the limited computing resources, we finally set the frame length to 16.

3) TCA numbers: The proposed method is composed of several TCA blocks, and it is necessary to analyze the effect of the number of TCA blocks on the final performance. We gradually increase the number of TCA blocks at intervals 2 shown in Table 6 (right).

As is mentioned above, there are two types of TCA blocks stacked alternately to make up the whole architecture. The difference between them is whether there exists downsampling operation. When changing the number of TCA blocks, we only add or remove blocks without downsampling. Therefore, the output size of the network is maintained at 1/16 of the input resolution. It is obvious that the network delivers the best performance when the number of blocks equals to 8.

5 Conclusion

In this paper, we propose a novel block named temporal channel-aware (TCA) block, which not only captures the temporal dependencies in video sequences, but also combines global context information with local spatio-temporal features to boost the accuracy for crowd counting. By stacking the TCA blocks to form the enhanced 3D convolutional network (E3D), we can achieve state-of-the-art performance over the existing methods on three benchmarks. Besides, we propose a degenerate variant of E3D by replacing 3D convolutions with 2D convolutions and test it on the vehicle datatset TRANCOS, which demonstrates our model can still achieve good results in case the temporal information is not available.

References

[1] Deepak Babu Sam, Shiv Surya, and R. Venkatesh Babu. Switching convolutional neural network for crowd counting. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[2] Deepak Babu Sam, Neeraj N. Sajjan, R. Venkatesh Babu, and Mukundhan Srinivasan. Divide and grow: Capturing huge diversity in crowd images with incrementally growing cnn. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[3] Xinkun Cao, Zhipeng Wang, Yanyun Zhao, and Fei Su. Scale aggregation network for accurate and efficient crowd counting. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[4] A. B. Chan and N. Vasconcelos. Bayesian poisson regression for crowd counting. In *2009 IEEE 12th International Conference on Computer Vision*, pages 545–551, Sep. 2009. doi: 10.1109/ICCV.2009.5459191.

[5] A. B. Chan, , and N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, June 2008. doi: 10.1109/CVPR.2008.4587569.

[6] Ke Chen, Chen Change Loy, Shaogang Gong, and Tony Xiang. Feature mining for localised crowd counting. In *BMVC*, 2012.

[7] Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In *International Conference on Computer Vision & Pattern Recognition (CVPR '05)*, volume 1, pages 886–893, June 2005. doi: 10.1109/CVPR.2005.177.

[8] Diptodip Deb and Jonathan Ventura. An aggregated multicolumn dilated convolution network for perspective-free counting. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[9] C.Fookes D.Ryan, S.Deman and S.Sridharan. Crowd counting using multiple local features. *Digital Image Computing:Techniques and Applications*, pages 81–88.IEEE, Feb 2009. doi: 2009.DICTA'09.

[10] Ricardo Guerrero-Gómez-Olmedo, Beatriz Torre-Jiménez, Roberto López-Sastre, Saturnino Maldonado-Bascón, and Daniel Oñoro-Rubio. Extremely overlapping vehicle counting. In Roberto Paredes, Jaime S. Cardoso, and Xosé M. Pardo, editors, *Pattern Recognition and Image Analysis*, pages 423–431, Cham, 2015. Springer International Publishing. ISBN 978-3-319-19390-8.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] D. Kang, Z. Ma, and A. B. Chan. Beyond counting: Comparisons of density maps for crowd analysis tasks - counting, detection, and tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1, 2018. ISSN 1051-8215. doi: 10.1109/TCSVT.2018.2837153.

[13] Di Kang and Antoni Chan. Crowd counting by adaptively fusing predictions from an image pyramid. In *The British Machine Vision Conference*, 2018.

[14] M. Li, Z. Zhang, K. Huang, and T. Tan. Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection. In *2008 19th International Conference on Pattern Recognition*, pages 1–4, Dec 2008. doi: 10.1109/ICPR.2008.4761705.

[15] Yuhong Li, Xiaofan Zhang, and Deming Chen. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. *CoRR*, abs/1802.10062, 2018.

[16] Jiang Liu, Chenqiang Gao, Deyu Meng, and Alexander G. Hauptmann. Decidenet: Counting varying density crowds through attention guided detection and density estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[17] Xialei Liu, Joost van de Weijer, and Andrew D. Bagdanov. Leveraging unlabeled data for crowd counting by learning to rank. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[18] Xin Miao, Xiantong Zhen, Xianglong Liu, Cheng Deng, Vassilis Athitsos, and Heng Huang. Direct shape regression networks for end-to-end face alignment. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[19] Daniel Oñoro-Rubio and Roberto J. López-Sastre. Towards perspective-free object counting with deep learning. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 615–629, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46478-7.

[20] Daniel Oñoro-Rubio, Mathias Niepert, and Roberto J López-Sastre. Learning short-cut connections for object counting. *arXiv preprint arXiv:1805.02919*, 2018.

[21] V. Pham, T. Kozakaya, O. Yamaguchi, and R. Okada. Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3253–3261, Dec 2015. doi: 10.1109/ICCV.2015.372.

[22] Viresh Ranjan, Hieu Le, and Minh Hoai. Iterative crowd counting. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[23] B. Sheng, C. Shen, G. Lin, J. Li, W. Yang, and C. Sun. Crowd counting via weighted vlad on a dense attribute feature map. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(8):1788–1797, Aug 2018. ISSN 1051-8215. doi: 10.1109/TCSVT.2016.2637379.

[24] Zenglin Shi, Le Zhang, Yun Liu, Xiaofeng Cao, Yangdong Ye, Ming-Ming Cheng, and Guoyan Zheng. Crowd counting with deep negative correlation learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[25] Vishwanath A. Sindagi and Vishal M. Patel. Generating high-quality crowd density maps using contextual pyramid cnns. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[26] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[27] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. *CoRR*, abs/1711.11248, 2017.

[28] Paul Viola and Michael J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, May 2004. ISSN 1573-1405. doi: 10.1023/B:VISI.0000013087.49260.fb.

[29] Y. Wang and Y. Zou. Fast visual object counting via example-based density estimation. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3653–3657, Sep. 2016. doi: 10.1109/ICIP.2016.7533041.

[30] Ze Wang, Zehao Xiao, Kai Xie, Qiang Qiu, Xiantong Zhen, and Xianbin Cao. In defense of single-column networks for crowd counting. *CoRR*, abs/1808.06133, 2018.

[31] Bo Wu and Ram Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, 75(2):247–266, Nov 2007. ISSN 1573-1405. doi: 10.1007/s11263-006-0027-7.

[32] Feng Xiong, Xingjian Shi, and Dit-Yan Yeung. Spatiotemporal modeling for crowd counting in videos. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[33] Cong Zhang, Hongsheng Li, Xiaogang Wang, and Xiaokang Yang. Cross-scene crowd counting via deep convolutional neural networks. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 833–841, June 2015. doi: 10.1109/CVPR.2015.7298684.

[34] Shanghang Zhang, Guanhang Wu, João P. Costeira, and José M. F. Moura. Fcn-rlstm: Deep spatio-temporal neural networks for vehicle counting in city cameras. *CoRR*, abs/1707.09476, 2017.

[35] Yiheng Zhang, Zhaofan Qiu, Ting Yao, Dong Liu, and Tao Mei. Fully convolutional adaptation networks for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[36] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[37] Z. Zou, X. Su, X. Qu, and P. Zhou. Da-net: Learning the fine-grained density distribution with deformation aggregation network. *IEEE Access*, 6:60745–60756, 2018. ISSN 2169-3536. doi: 10.1109/ACCESS.2018.2875495.