17th International Conference on Machine Vision Applications (MVA)
Fully Online, July 25–27, 2021.

Temporal Extension for Encoder-Decoder-based Crowd Counting Approaches

Thomas Golda[1,2,*] Florian Krüger[2,*] Jürgen Beyerer[1,2]

[1]Vision and Fusion Laboratory, Karlsruhe Institute of Technology KIT
[2]Fraunhofer Institute for Optronics, System Technologies and Image Exploitation IOSB
Fraunhofer Center for Machine Learning
firstname.lastname@iosb.fraunhofer.de

Abstract

Crowd counting is an important aspect to safety monitoring at mass events and can be used to initiate safety measures in time. State-of-the-art encoder-decoder architectures are able to estimate the number of people in a scene precisely. However, since most of the proposed methods are based to solely operate on single-image features, we observe that estimated counts for aerial video sequences are inherently noisy, which in turn reduces the significance of the overall estimates. In this paper, we propose a simple temporal extension to said encoder-decoder architectures that incorporates local context from multiple frames into the estimation process. By applying the temporal extension a state-of-the-art architectures and exploring multiple configuration settings, we find that the resulting estimates are more precise and smoother over time.

1 Introduction

With the rapid spread of COVID-19 early in 2020, the year became another memorable moment in time that led to drastic changes in public life showing the importance of global solidarity. Alongside strong restrictions in local public transport, shops and public places like parks or malls, almost all recurring public events including fairs were cancelled. Especially public events that allure thousands and thousands of people will have to deal with upcoming consequences of such a situation. *Social distancing* is just one phrase that gained popularity within recent time. However, having an overview over visitors of such huge events did not just come up during the COVID-19 pandemic. Especially from a public safety view, organizers want to have an overview of how crowds of people distribute over a certain area and how they move in particular since classical public events are growing in count and size. For such events, this is not just interesting from a monitoring point of view but also for evacuation simulation at a preliminary stage. Especially for major events it is common to hire companies that perform appropriate simulations in order to analyze their safety and evacuation concepts. Since simulations like these are based on real data, collecting sufficient and proper data is inevitable. Although, hiring staff to use tally counter in order to determine the number of people entering or leaving the event area, is a time and resource consuming way prone to errors due to the limited amount of attention of single persons. This brings up the notion of an automatic evaluation, which comes with further benefits mainly driven through the inexhaustibility of machines and even more due to their ability to easily work on an holistic level. While in particular static public events with reserved areas have access to pre-installed video cameras, smaller and emerging events typically do not have those devices at their disposal. Drones however are becoming more and more inexpensive and flexible and therefore are an appropriate way to collect data that can easily be used for simulation processes.

In this work we will first give an overview over related work, including classical ground-based crowd counting work as well as those done from aerial imagery collected from drones, helicopters or even with wide-area motion imagery sensors. This is supplemented by a short summary of existing datasets suitable for development and evaluation of such algorithms. In the subsequent sections we will present our proposed approach to tackle the problem of crowd analysis from aerial-collected video footage, followed by a thorough evaluation on suitable datasets.

2 Related Work

Crowd counting in single images has been the main focus of research so far, where mainly the case of *perspective* imagery has been covered [1, 2]. CNN-based approaches based on encoder-decoder architectures that regress on a *density map* form the state-of-the-art here. Since crowd counting is closely related to the field of *semantic segmentation*, techniques from the latter were adopted and have been applied successfully, i.e. the use of dilated convolutions [3] and skip

*Both authors contributed equally to this work.

connections to forward different intermediate feature representations from the encoder into the decoder [4]. The latter was also proposed for the case of *aerial imagery* from bird's-eye view [5], in which individuals are depicted by only a few pixels.

Crowd counting in video sequences has thus far not been studied as extensively as the single-image case. In [6], a crowd counting architecture is constructed from a *convolutional* LSTM (ConvLSTM), followed by a single 1×1 convolutional layer. A VGG-16 feature extractor [7] is used in [8] to add intermediate feature maps from the current and one previous frame, both of which are apart by a fixed distance. The estimated density map for the current frame is then produced in decoder fashion. Lastly, the approach in [9] captures temporal dependencies at density map level within a centered window around the current frame. Weights for each density map within the window are derived, which are then used to combine them *convexly* to obtain the final density map for the current frame.

3 Temporal Extension

3.1 ConvLSTM

ConvLSTMs were proposed in [10] to exploit spatio-temporal dependencies in grid structured input data. We adapt them to operate fully convolutional by omitting terms in which additional fixed-size matrices are multiplied element-wise with the previous cell state. Filter kernels are assumed to be square and of same size. Furthermore, we adopt the approach of [11] to use ReLU for the non-linear activation function ϕ, in order to intersperse the ConvLSTM in between convolutional layers of a crowd counter.

3.2 Architecture

Extending single-frame architectures to incorporate temporal input dependencies is a common practice in the related field of semantic segmentation [12, 13], hence we apply it similarly to crowd counting. To be more specific, a ConvLSTM is included in between encoder and decoder. The former extracts a variety of local features from the current input frame $\boldsymbol{F}_t$, which are subsequently decoded by the latter into an estimated density map $\hat{D}_t$. In the following, the output of the encoder $\boldsymbol{E}_t := \text{Encoder}(\boldsymbol{F}_t)$ is referred to as the *embedding* of the current frame.

3.2.1 Learning Task

The temporal context consists of k embeddings comprising $\boldsymbol{E}_{t-k+1}$ up to and including $\boldsymbol{E}_t$. The estimated density map $\hat{D}_t$ for the current frame is then obtained using the final hidden state $\boldsymbol{h}_t$ of the ConvLSTM (Eq. 1).

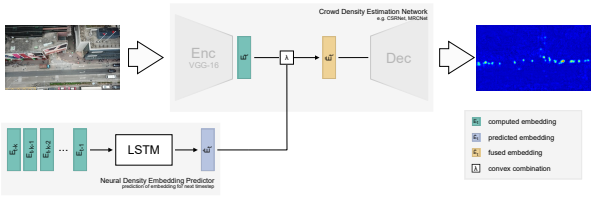


Figure 1: Visualization of the M2O architecture and complete pipeline. Both, the current embedding $\boldsymbol{E}_t$ and the predicted $\hat{\boldsymbol{E}}_t$ get fused to generate $\tilde{\boldsymbol{E}}_t$ as input for the decoder.

$$\begin{aligned} \hat{\boldsymbol{E}}_t &:= \boldsymbol{h}_t = \text{ConvLSTM}(\boldsymbol{E}_{t-k+1}, \ldots, \boldsymbol{E}_t) \\ \hat{D}_t &:= \text{Decoder}(\hat{\boldsymbol{E}}_t) \end{aligned} \tag{1}$$

As such, the *many-to-one* (M2O) learning task is considered. Note that the processing time of a single density map increases here roughly by k. Since the problem of crowd counting in video sequences deals with input-output sequences of equal length, we can also configure the ConvLSTM such that each of its k resulting hidden states are being decoded, hence tackling the *many-to-many* (M2M) learning task.

3.2.2 Merging Embeddings

With the ConvLSTM acting as an intermediate layer, it is responsible for two tasks. It extracts local temporal dependencies from the input and then merges them with the embedding into a suited semantic representation for decoding. Motivated by the fact that single-frame architectures already achieve good results on their own, only the former task is to be tackled by the ConvLSTM. We therefore propose to *merge* hidden states and embeddings by means of a convex combination with weight $\lambda \in [0, 1]$, prior to decoding (Eq. 2).

$$\hat{D}_t := \text{Decoder}(\lambda \cdot \boldsymbol{E}_t + (1 - \lambda) \cdot \hat{\boldsymbol{E}}_t) \tag{2}$$

The weight λ can either be fixed or set to be a learnable parameter. Here it is necessary to use ReLU for ϕ, as otherwise the semantic representations of $\boldsymbol{E}_t$ and $\hat{\boldsymbol{E}}_t$ would differ.

4 Experiments

4.1 Setup

All models were implemented in *PyTorch* (v1.4) and trained using a single Nvidia GeForce RTX 2080 Ti GPU with 11 GB of memory. We used *Adam* [14] for optimization at default settings to minimize the mean squared error. All models are trained until convergence of the training loss and the weights performing best on the validation set are used. Note that images were inferred as a whole.

4.1.1 Drone Crowd Dataset

Crowd counting in aerial imagery has only gained attention recently. To the best of our knowledge, the *Drone Crowd* dataset (DCD) [8] is the only available dataset comprising aerial video sequences. It consists of 82 sequences of 30 frames, which were recorded at 1,920 × 1,080 pixels and at an estimated frame rate of 1 FPS. The counts range from 25 to 421 persons. In order to generate ground truth density maps, we follow the approach of [5] to generate fixed size kernels using the *ground sampling distance* of an image. The latter is not given, therefore we measured the head size of a person in pixels and assumed an average head diameter of roughly 0.17 meters [15].

Since the official test set does not provide annotations, we created[1] one randomly from 16 sequences. For training, we randomly crop 20 sequences of size 256 × 256 pixels from each sequence. Furthermore, we augment these subsequences randomly with scaling between 0.5 and 2, rotation by multiples of 90° and flipping in both directions. Besides that, we also applied photometric distortions [16]. Training sequences of fixed size are obtained from these subsequences in a sliding window manner such that each frame is the final frame once within a training sequence.

4.1.2 Metrics

The crowd counting performance of a model is commonly measured by the absolute deviation of estimated counts $\hat{C}_0, \ldots, \hat{C}_{n-1}$, with $\hat{C}_i := \sum_x \sum_y \hat{D}_i(x, y)$. This is considered by both *mean absolute error* (MAE) and *root mean squared error* (RMSE), respectively [17].

Furthermore, for sequences of consecutive counts $\hat{C}_{i,0}, \ldots, \hat{C}_{i,m-1}$, we introduce a third metric to measure the deviation in *roughness* for all l videos, which we refer to as *mean absolute roughness error* (MARE) (Eq. 4). The roughness ρ_i (Eq. 3) for the i-th sequence is given by the standard deviation of successive count differences $\Delta_{i,j} := C_{i,j+1} - C_{i,j}$. It measures the average deviation from the overall trend of the progression.

$$\rho_i^2 := \frac{1}{m-1} \sum_{j=0}^{m-2} \left(\Delta_{i,j} - \overline{\Delta}_i\right)^2 \tag{3}$$

$$\text{MARE} := \frac{1}{l} \sum_{i=0}^{l-1} |\hat{\rho}_i - \rho_i| \tag{4}$$

4.2 Baseline

Although there are many state-of-the-art encoder-decoder architectures to cope with perspective imagery, not as many have been proposed for the case of aerial imagery. Fortunately, it appears as if the former can also be applied here successfully, judging by the *VisDrone2020 Crowd Counting Challenge* [18]. Therefore we set CSRNet [3] and SFANet [4] as our baselines, as they performed well therein. We also include MRCNet [5], since it was proposed for crowd counting in aerial imagery. Note that for reasons of comparison, SFANet is employed without the attention-map decoder.

Table 1: Evaluation of single-frame baseline architectures. Batch sizes and learning rates were set to (40, $5 \cdot 10^{-6}$) (CSRNet), (20, 10^{-5}) (MRCNet) and (40, 10^{-5}) (SFANet). We set the loss parameter $\lambda = 0.1$ during training of the MRCNet.

Method	MAE	RMSE	MARE
CSRNet	**35.5**	**43.4**	3.16
MRCNet	46.7	58.3	3.19
SFANet	39.7	48.3	**2.74**

We observe from Tab. 1 that CSRNet performs the best in terms of crowd counting, whereas the deviation in roughness of estimated count progressions is lowest for SFANet. Fluctuations in estimated counts over time are particularly observed for moving individuals, which causes their depicted shapes to vary from frame to frame.

4.3 Temporal Extension

The temporal extension is included after the final 512 channel convolutional layer. Since the decoder of CSRNet is constructed from dilated convolutions with dilation rate of 2, we also adopt them in the ConvLSTM with 3 × 3 filters. We evaluate our approach for contexts of three and five embeddings, i.e. $k \in \{3, 5\}$. Larger contexts were not considered due to limited training time and the low frame rate of sequences in DCD.

4.3.1 Learning Task

At first, the temporal extension is evaluated for both learning tasks and context sizes. As shown in Tab. 2,

Table 2: Evaluation results of the temporal extension (TE) with both learning tasks.

	Method	MAE	RMSE	MARE
$k = 3$	TE-M2O	**31.5**	**39.8**	2.61
	TE-M2M	34.4	42.5	2.87
$k = 5$	TE-M2O	42.2	53.3	**2.03**
	TE-M2M	38.9	49.4	3.93

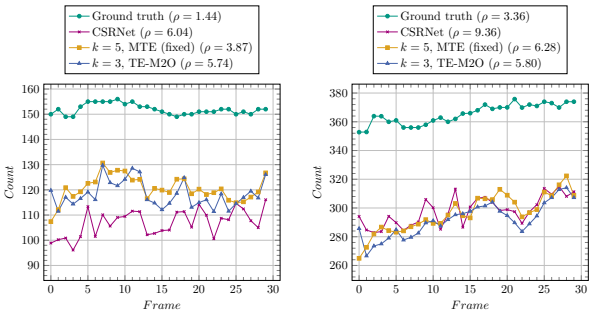
[1] *Validation*: 28-30, 36-38, 51-53, 83, 85, 92, 93, 102; *Test*: 13, 14, 19-21, 39-41, 54, 71-73, 78, 79, 91, 110

Table 3: Evaluation results of the temporal extension with embedding merging (MTE). Weight λ was either fixed to 0.5 or set to be a learnable parameter.

	Method	MAE	RMSE	MARE
$k = 3$	MTE (fixed)	41.1	51.5	**2.08**
	MTE (learned)	41.6	52.7	2.24
$k = 5$	MTE (fixed)	**31.9**	**38.7**	2.17
	MTE (learned)	33.6	43.6	2.19

Table 4: Evaluation results of MRCNet and SFANet with configurations TE-M2O ($k = 3$) and MTE ($k = 5$) utilizing three and five embeddings, respectively.

Method	Arch.	MAE	RMSE	MARE
TE-M2O	MRCNet	46.7	59.8	2.05
	SFANet	46.0	55.5	2.35
MTE (fixed)	MRCNet	44.3	56.9	**1.58**
	SFANet	**33.2**	**41.8**	1.82



(a) Sequence S_{73}. (b) Sequence S_{110}.

Figure 2: Exemplary estimated count progressions for two sequences of the test set. The temporal extension was used with CSRNet. Both sequences are equally challenging for all methods. S_{73} shows a dark scene with little contrast, S_{110} on the other hand shows a road junction with lots of structure.

the largest improvement in terms of crowd counting performance is achieved when using three embeddings and the M2O task, where the MAE is about 11% lower. The performance degrades with five embeddings, although it yields the smoothest progression.

4.3.2 Merging Embeddings

Although it cannot be said which of the tasks is superior, we evaluate the approach of merging embeddings using the M2O task. With different context sizes, it performed best in terms of crowd counting performance and smoothness of estimated count progressions.

The results of the embedding-merging approach are reported in Tab. 3. At first, we fixed λ at 0.5 and achieved the best crowd counting performance when using five embeddings. Although the MAE increased by about 1.3% compared to the previous approach, the resulting RSME could be lowered by 2.8%. This shows a larger impact of this change on strongly deviating count estimates. The by 0.44 reduced MARE shows smoother count progressions on these estimates. The setup yielding smoothest estimates was using three embeddings, but decreased the crowd counting performance on the counterside.

Setting λ to be learnable, we expected the performance in both categories to at least not degrade, since the network should be able to find a suited weight. However, the results of Tab. 3 report otherwise, as performance degrades in both categories and for both context sizes. The final learned values for λ were 0.47 ($k = 3$) and 0.33 ($k = 5$), where the network tends to favor embeddings $\boldsymbol{E}_t$. However, the shape of progressions of λ during training were the same, irrespective of the context size and initial value. It appears as if the addition of the extra parameter does not introduce extra dynamic into the optimization problem.

We also applied the top two performing configurations of the temporal extension on the remaining crowd counters besides CSRNet. Similarly, a batch size of 10 was used and the learning rate of the respective crowd counter was adjusted adequately. The results are reported in Tab. 4. With MTE (fixed), we observed improved crowd counting performance and smoother estimated progressions for both crowd counters. Therefore, it seems as if this architecture is the better approach to crowd counting in video sequences. In contrast, only a decrease in MARE is the case when using TE-M2O, while an improvement in terms of crowd counting performance cannot be observed.

Finally, Fig. 2 depicts plots of estimated count progressions for models TE-M2O and MTE (fixed) (both using CSRNet) utilizing three and five embeddings, respectively. Estimates up until the second and fourth frame, respectively, are somewhat noisy, which is caused from padding the beginning of some subsequences with empty frames.

5 Conclusion

In this paper, we proposed to include a ConvLSTM between encoder and decoder to enhance single-frame crowd counting architectures for tackling video sequences. We found that by also utilizing multiple previous embeddings, counts can be estimated more accurately, which are also smoother over time. By treating the feature extraction from the current frame and the capturing of temporal dependencies as two separate tasks, we furthermore observed that estimated count progressions were even smoother while the counts re-

main accurate. It remains to examine the effectiveness of different configurations of the input window that is fed into the ConvLSTM, such as a centered one as in [9], or also utilizing future frames in a bidirectional manner as in [6].

Acknowledgment

This research was funded by the German Federal Ministry of Education and Research (BMBF) grant number 13N15164 (ESCAPE).

References

[1] Lokesh Boominathan, Srinivas S. S. Kruthiventi, and R. Venkatesh Babu, "Crowdnet: A deep convolutional network for dense crowd counting," *CoRR*, vol. abs/1608.06197, 2016.

[2] Chuan Wang, Hua Zhang, Liang Yang, Si Liu, and Xiaochun Cao, "Deep people counting in extremely dense crowds," in *Proceedings of the 23rd ACM International Conference on Multimedia*, 10 2015, pp. 1299–1302.

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

[4] Liang Zhu, Zhijian Zhao, Chao Lu, Yining Lin, Yao Peng, and Tangren Yao, "Dual path multi-scale fusion networks with attention for crowd counting," *CoRR*, vol. abs/1902.01115, 2019.

[5] R. Bahmanyar, Elenora Vig, and P. Reinartz, "Mrcnet: Crowd counting and density map estimation in aerial and ground imagery," *ArXiv*, vol. abs/1909.12743, 2019.

[6] Feng Xiong, Xingjian Shi, and Dit-Yan Yeung, "Spatiotemporal modeling for crowd counting in videos," *CoRR*, vol. abs/1707.07890, 2017.

[7] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun, Eds., 2015.

[8] Longyin Wen, Dawei Du, Peng-Fei Zhu, Q. Hu, Qilong Wang, L. Bo, and S. Lyu, "Drone-based joint density map estimation, localization and tracking with space-time multi-scale attention network," *ArXiv*, vol. abs/1912.01811, 2019.

[9] Xingjiao Wu, Baohan Xu, Yingbin Zheng, Hao Ye, Jing Yang, and Liang He, "Video crowd counting via dynamic temporal modeling," *CoRR*, vol. abs/1907.02198, 2019.

[10] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," *CoRR*, vol. abs/1506.04214, 2015.

[11] Mason Liu and Menglong Zhu, "Mobile video object detection with temporally-aware feature maps," *CoRR*, vol. abs/1711.06368, 2017.

[12] Qin Zou, Hanwen Jiang, Qiyu Dai, Yuanhao Yue, Long Chen, and Qian Wang, "Robust lane detection from continuous driving scenes using deep neural networks," *CoRR*, vol. abs/1903.02193, 2019.

[13] Seyed Shahabeddin Nabavi, Mrigank Rochan, and Yang Wang, "Future semantic segmentation with convolutional LSTM," *CoRR*, vol. abs/1807.07946, 2018.

[14] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," 2017.

[15] K M Bushby, T Cole, J N Matthews, and J A Goodship, "Centiles for adult head circumference.," *Archives of Disease in Childhood*, vol. 67, no. 10, pp. 1286–1287, 1992.

[16] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin, "MMDetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.

[17] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 589–597.

[18] Dawei Du, Longyin Wen, Pengfei Zhu, Heng Fan, Qinghua Hu, Haibin Ling, Mubarak Shah, Junwen Pan, Ali Al-Ali, Amr Mohamed, Bakour Imene, Bin Dong, Binyu Zhang, Bouchali Hadia Nesma, Chenfeng Xu, Chenzhen Duan, Ciro Castiello, Corrado Mencar, Dingkang Liang, Florian Krüger, Gennaro Vessio, Giovanna Castellano, Jieru Wang, Junyu Gao, Khalid Abualsaud, Laihui Ding, Lei Zhao, Marco Cianciotta, Muhammad Saqib, Noor Almaadeed, Omar Elharrouss, Pei Lyu, Qi Wang, Shidong Liu, Shuang Qiu, Siyang Pan, Somaya Al-Maadeed, Sultan Daud Khan, Tamer Khattab, Tao Han, Thomas Golda, Wei Xu, Xiang Bai, Xiaoqing Xu, Xuelong Li, Yanyun Zhao, Ye Tian, Yingnan Lin, Yongchao Xu, Yuehan Yao, Zhenyu Xu, Zhijian Zhao, Zhipeng Luo, Zhiwei Wei, and Zhiyuan Zhao, "Visdrone-cc2020: The vision meets drone crowd counting challenge results," in *Computer Vision – ECCV 2020 Workshops*, Adrien Bartoli and Andrea Fusiello, Eds., Cham, 2020, pp. 675–691, Springer International Publishing.