Repetitive Action Counting with Motion Feature Learning (Supplementary Material)

Xinjie Li, Huijuan Xu
Pennsylvania State University
University Park, USA
xql5497@psu.edu, hkx5063@psu.edu

1. More Ablation Analysis

Matrix Reconstruction. Here, we discuss the importance of matrix reconstruction in our proposed method and describe the specific training strategy we employ to enhance its effectiveness. As mentioned earlier in Section 3.3, the video reconstruction process relies on fixed target features, denoted as $\mathbf{Z_T}$ and $\mathbf{Z_T^F}$, which serve as templates for guiding the reconstruction. This fixed target feature approach is crucial for ensuring that the reconstruction is meaningful and does not introduce random artifacts that may adversely affect the temporal self-similarity matrices $\mathbf{S_B}$ and $\mathbf{S_B^F}$.

To address this, we adopt a training strategy where we freeze the modules preceding the temporal self-similarity (TSM) modules, as illustrated in Fig. 3, during the training phase that involves the similarity-based video reconstruction loss. Specifically, we first train the entire network, including all modules, using two prediction losses, denoted as $\mathbf{L_P}$ and $\mathbf{L_P^F}$. Once this initial training is complete, we then freeze the front modules and solely focus on training the TSM modules and their subsequent modules. This training phase incorporates the prediction losses $\mathbf{L_P}$ and $\mathbf{L_P^F}$, as well as the reconstruction losses $\mathbf{L_R}$ and $\mathbf{L_R^F}$.

To validate our hypothesis regarding the effectiveness of this training strategy, we conduct an experiment where we train the entire network using all prediction and reconstruction losses simultaneously. However, the performance achieved in this case is suboptimal, with an MAE of only 0.5187 and an OBO of 0.2649. In contrast, our proposed method, with the selective freezing and separate training of modules, achieves significantly better results, with an MAE of 0.3841 and an OBO of 0.3860.

These experimental findings clearly demonstrate that our design and training strategy significantly contribute to the effectiveness of our method. By selectively freezing modules and focusing on separate training phases, we can improve the performance of the network and achieve superior accuracy in repetitive action counting tasks.

Mask Rate. To evaluate the effectiveness of the masked matrix reconstruction method, we conduct an ablation study using different mask rates. We experiment with various mask rates and analyze their impact on the performance of our method. From Table 1, our analysis reveals that a mask rate of 0.25 yields the best performance among the tested rates.

Mask Rate	MAE↓	OBO ↑
0.75	0.3971	0.3371
0.5	0.3956	0.3509
0.25	0.3909	0.3576

Table 1. Performance for different mask rates in the masked matrix reconstruction method. The mask rate represents the division between the number of frames that are masked and the total number of frames.

Variance-Integrated Loss Weights Generation *vs.* Handcrafted Loss Weights Design. We compare the performance of our proposed variance-integrated loss weights generation approach with the traditional handcrafted loss weights design. Table 2 presents the performance metrics obtained by both methods, highlighting the superiority of our proposed approach. The proposed approach leverages the inherent variance of the training data to dynamically adjust the loss weights, allowing for a more adaptive and precise optimization process. This results in improved accuracy and better convergence of the model during training. In contrast, the handcrafted loss weights design relies on manual tuning and predefined weight values. While this approach can achieve reasonable results, it lacks the adaptability and flexibility of our proposed method.

Our experimental results demonstrate the effectiveness and efficiency of the variance-integrated loss weights generation method, which automatically adapts the loss weights based on the data's inherent characteristics. This approach eliminates the need for manual fine-tuning and provides a more robust and scalable solution for the repetitive action

RGB:FLow	MAE↓	OBO ↑
0.2:0.8	0.3926	0.3642
0.4:0.6	0.3924	0.3779
0.5:0.5	0.3909	0.3576
0.6:0.4	0.3969	0.3642
0.8:0.2	0.3847	0.3709
Ours	**0.3841**	**0.3860**

Table 2. Performance comparison of variance-integrated loss weights generation and handcrafted loss weights design.

counting task.

2. Predicted Density Map Visualization

In this section, we present visualizations of the predicted density maps generated by our method. Figure 1 illustrates the predicted density maps generated by our method. It is evident that our approach effectively handles abrupt background changes and produces accurate density maps for action counting. This demonstrates the robustness and reliability of our method in capturing and quantifying repetitive actions in various environmental settings.

However, as depicted in Figure 2, our method encounters difficulties in handling long breaks during the video. These interruptions in the action sequence pose a challenge for accurate counting. We acknowledge that this is an area for improvement and will be the focus of our future work.

3. Baseline Evaluation on UCFRep Dataset

To enable a comprehensive performance comparison on the UCFRep dataset [7], we carefully review the relevant literature [3, 7, 8]. Our examination reveals that [3] does not conduct experiments on the UCFRep dataset, while [8] does not provide code related to the UCFRep dataset. Additionally, although the code for [7] is available, reproducing their reported results has proven challenging, as confirmed by other researchers [1], and has also been noted in another recent work [4].

To address these limitations and ensure a fair comparison, we have re-implemented these methods specifically for the UCFRep dataset [7]. For consistency, we employ the 3D-ResNext101 model [2] pre-trained on Kinetics [1] as our encoder and maintain a consistent input size of 112×112, following the settings of [7]. It is important to note that we construct 64 frames for each video and conduct our experiments without any data augmentation. Furthermore, we have re-implemented [8] with the S3D encoder [6] and [3] with the Video Swin Transformer Tiny encoder [5],

[1] https://github.com/Xiaodomgdomg/Deep-Temporal-Repetition-Counting/issues/8

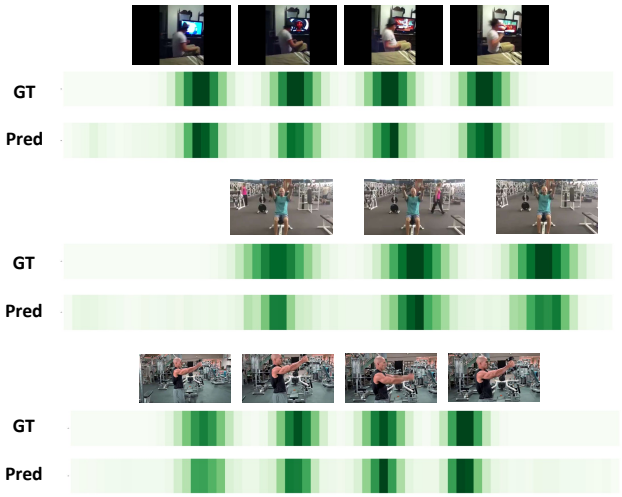

Figure 1. Visualization of predicted density maps by our method showcases its ability to handle abrupt background changes and accurately predict density maps for counting repetitive actions. All three videos contain abrupt background changes, which are the television screen, moving person, and moving camera from top to bottom. The illustration examples are from the RepCount dataset [3].

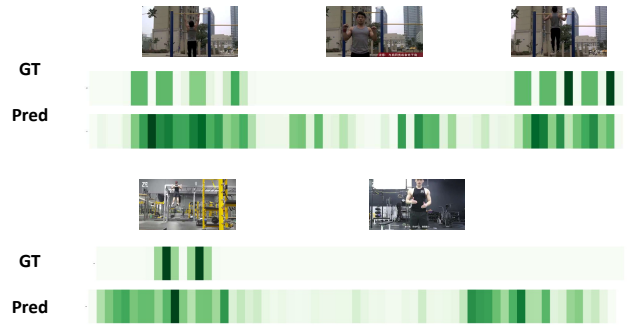

Figure 2. Illustration of the influence of long breaks during the video (the men stop to explain the exercise) on our method's performance, highlighting an area for improvement in future work. The illustration examples are from the RepCount dataset [3].

in line with the specifications outlined in their respective papers. We have meticulously followed the implementation details provided in the original papers [3, 7, 8] to ensure a fair and accurate comparison.

Table 3 presents the results of our baseline evaluation on the UCFRep dataset. It is evident that our re-implemented method achieves state-of-the-art performance, surpassing all other baseline methods. This demonstrates the effectiveness and superiority of our approach in the task of repetitive action counting on the UCFRep dataset.

Method	Encoder	MAE↓	OBO ↑
⋆ Context [7]	3D-ResNext101 [2]	0.1470	0.7900
⋆ Zhang *et al*. [8]	S3D [6]	0.1430	0.8000
#Context [7]	N/A	0.7620	0.4120
#TransRAC [3]	N/A	0.6401	0.3240
† Zhang *et al*. [8]	3D-ResNext101 [2]	0.4825	0.3125
† Zhang *et al*. [8]	S3D [6]	0.4129	0.3542
† Context [7]	3D-ResNext101 [2]	0.4689	0.4800
† TransRAC [3]	3D-ResNext101 [2]	0.4409	0.4300
† TransRAC [3]	Video Swin Tiny [5]	0.4139	0.4200
Ours	3D-ResNext101 [2]	**0.3879**	**0.5100**

Table 3. Performance comparison of various baseline methods on the UCFRep dataset [7], demonstrating the state-of-the-art performance achieved by our re-implemented method. † Our re-implemented results. ⋆ Results from corresponding works. # Results from Full [4].

References

[1] João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 4724–4733. IEEE Computer Society, 2017. 2

[2] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet. *computer vision and pattern recognition*, 2017. 2, 3

[3] Huazhang Hu, Sixun Dong, Yiqun Zhao, Dongze Lian, Zhengxin Li, and Shenghua Gao. Transrac: Encoding multi-scale temporal correlation with transformers for repetitive action counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19013–19022, 2022. 2, 3

[4] Jianing Li, Bowen Chen, Zhiyong Wang, and Honghai Liu. Full resolution repetition counting. *arXiv preprint arXiv:2305.13778*, 2023. 2, 3

[5] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *arXiv: Computer Vision and Pattern Recognition*, 2021. 2, 3

[6] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. *Cornell University - arXiv*, 2017. 2, 3

[7] Huaidong Zhang, Xuemiao Xu, Guoqiang Han, and Shengfeng He. Context-aware and scale-insensitive temporal repetition counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 670–678, 2020. 2, 3

[8] Yunhua Zhang, Ling Shao, and Cees GM Snoek. Repetitive activity counting by sight and sound. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14070–14079, 2021. 2, 3