Advanced Video Inpainting method using Residual Query Connection

Youngjun La[1[0000−0002−0426−7939]] and Jong-Il Park[1[0000−0003−1000−4067]]

Hanyang University, Department of Computer Science, Republic of Korea
{yjla,jipark}@hanyang.ac.kr

Abstract. In this paper, we propose a method to enhance the performance of video inpainting using Residual Query Connection. Video inpainting is a method of visually filling in the damaged regions in each video frame. Recently, Transformer-based video inpainting has shown remarkable performance, however, it has a drawback of slow model speed when using video as input. A simple way to increase the model speed is to decrease the number of Transformer blocks used. Our proposed method adds local feature information by performing Multi-head Self-Attention within the residual connection of the Query. As evidenced by 2% and 2.6% improvement in LPIPS and FID, our approach is shown to slightly improve degraded performance by reducing the number of Transformer blocks. Additionally, we measure performance per 100K iterations and conduct qualitative evaluations, demonstrating the effectiveness of our proposed method.

Keywords: Computer Vision · Deep Learning · Video Inpainting · Vision Transformer.

1 Introduction

Video inpainting is a method of filling in damaged areas in each video frame in a realistic manner. Video inpainting is widely used in various fields such as video completion and object removal [11]. Although image inpainting has greatly advanced, performing inpainting on each frame can result in inconsistent inpainting results [12, 17, 18]. Recently, deep learning-based video inpainting research has been actively conducted to achieve temporal consistency [3, 19, 6, 7, 5, 2].

Among them, the optical flow-based video inpainting model applies pixel propagation to achieve temporal consistency [16, 3]. However, this pixel propagation requires manual operations such as Poisson blending, making it slow. To solve this, E^{2}FGVI (End-to-End Framework for Flow-Guided Video Inpainting) modifies the model structure to enable feature propagation instead of pixel propagation, resulting in improved video inpainting speed [9]. However, E^{2}FGVI still has a slow speed of 0.095 seconds per frame on an NVIDIA GeForce RTX 2080ti GPU. A simple way to increase the speed of this model is to reduce the number of Transformer blocks used in the model. By reducing the number of blocks from 8 to 4, the model speed increases to 0.068 seconds per frame. However, this approach reduces the depth of the model and therefore the inpainting performance decreases.

This paper proposes Residual Query Connection to improve the degraded inpainting performance. The proposed method adds local feature information by connecting the extracted query from the local window to the residuals during the Multi-head Self-Attention of E^{2}FGVI. This approach can improve the inpainting performance that has been degraded by reducing the number of Transformer blocks in the existing model. Furthermore, by conducting qualitative evaluations, it is shown that the proposed method is effective in diverse damaged video scenes generated by the DAVIS dataset, thus it can be applied in various fields.

The sequence of this paper is as follows: In Section 2, E^{2}FGVI model structure, which is used as the basic model of the proposed method, is introduced. In Section 3, the proposed method, Residual Query Connection, is explained. In Section 4, the experimental method is described, in Section 5, the results and analysis of the experiment are presented. Finally, in Section 6, the conclusion and future research directions of this paper are presented.

2 E^{2}FGVI

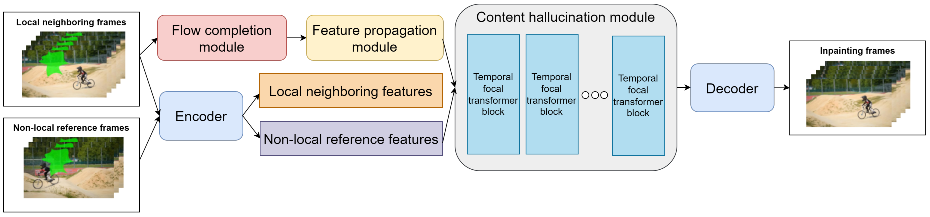
Local neighboring frames
Non-local reference frames
Flow completion module
Feature propagation module
Encoder
Local neighboring features
Non-local reference features
Content hallucination module
Temporal focal transformer block
Temporal focal transformer block
Temporal focal transformer block
Decoder
Inpainting frames

Fig. 1. E^{2}FGVI Model Structure

E^{2}FGVI (End-to-End Framework for Flow-Guided Video Inpainting) is an end-to-end video inpainting model in which modules are performed in the order of flow completion, feature propagation, and content hallucination [9]. The model structure is shown in Figure 1.

The components of the proposed model are as follows. First, the encoder is used to lower the resolution of the features for computational efficiency. Next, the flow completion module is used to complete the optical flow of the local neighboring frames. The flow completion module uses the lightweight model SPyNet [14]. Then, the feature propagation module is used to bidirectionally propagate the local neighboring features based on the completed optical flow. The feature propagation module uses deformable convolution [21]. Next, the multi-layer temporal focal transformer is used to perform content hallucination. The temporal focal transformer is an extension of the Focal Transformer for videos, which improves the interaction between local and global features [10]. Finally, the decoder is used to increase the size of the features and output the video inpainting results. This model serves as the base model for the proposed method.

3 Residual Query Connection

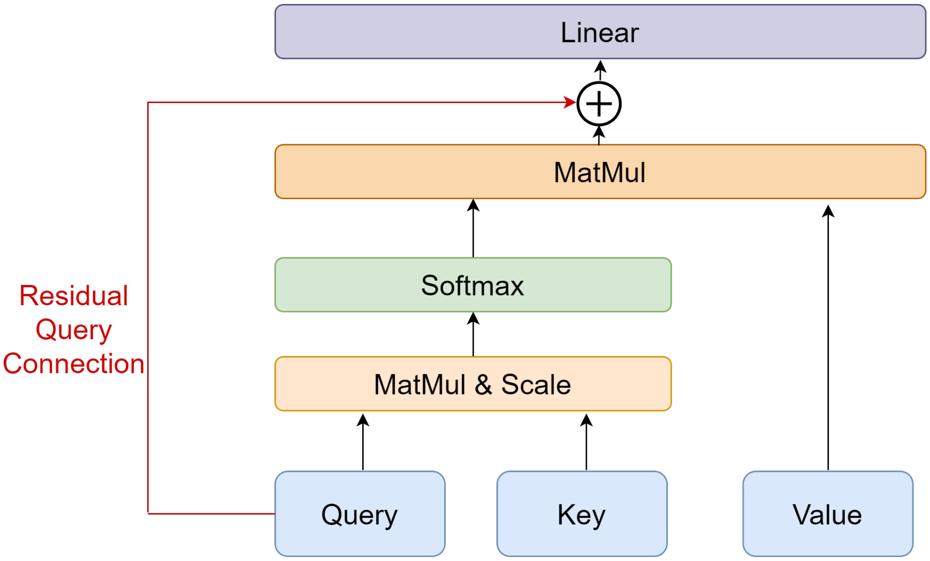

Fig. 2. Multi-head Self-Attention with Residual Query Connection

Residual Query Connection is a method for connecting the residual of the Query in Multi-head Self-Attention of the base model's temporal focal transformer. As shown in Figure 2, the proposed method adds Residual Query Connection to the existing Multi-head Self-Attention. This method is applied by modifying the Residual pooling connection used in previous studies [8]. The difference between ours and previous is that the previous research was used in pooling attention to combine local features before attention, while this research used it in window attention. Specifically, our method connects the entire Query, not the Query that has been pooled. This method has the effect of adding additional local feature information of the Query. This improves the inpainting performance for damaged videos that require more local feature information.

4 Experimental method

In this paper, we conduct the training in the same way as previous studies that used the E^2FGVI model [9]. The difference is that our proposed method is added, and to increase the model speed, we reduce the number of Transformer blocks used in the previous study from 8 to 4. The training data used is the video object segmentation dataset YouTube-VOS with 3471 samples [15]. Commonly, the batch size is 7, and the learning rate starts at 0.0001 and is reduced by a factor of 10 when the number of iterations reaches 400K. The optimizer used is Adam with β_1=0 and β_2=0.99. The total number of iterations is 500K, and we use 4 NVIDIA GeForce RTX 2080ti GPUs for training. The model's input consists of 5 consecutive local neighboring frames and 3 randomly extracted global reference frames. The resolution of each video frame is 432x240. The loss function used is the same as previous studies, which is the optical flow loss function L_{flow} and the loss function L_{rec} that calculates the pixel-wise difference between the inpainted video $\hat{Y}$ and the original video Y using the L1 distance [9]. Additionally, we use the loss function L_D of the discriminator that focuses on local and global features between neighboring frames and the adversarial loss function L_{adv} of the generator [1]. The weights of L_{flow}, L_{rec}, and L_{adv} were 1, 1, and 10^{-2}, respectively.

$$L_{flow} = \sum_{t=1}^{T-1} \left\| \hat{F}_{t\rightarrow t+1} - F_{t\rightarrow t+1} \right\|_1 + \sum_{t=2}^{T} \left\| \hat{F}_{t\rightarrow t-1} - F_{t\rightarrow t-1} \right\|_1 \tag{1}$$

$$L_{rec} = \left\| \hat{Y} - Y \right\|_1 \tag{2}$$

$$L_D = E_{x\sim P_Y(x)}[ReLU(1-D(x))] + E_{z\sim P_{\hat{Y}}(z)}[ReLU(1+D(z))] \tag{3}$$

$$L_{adv} = -E_{z\sim P_{\hat{Y}}(x)}[D(z)] \tag{4}$$

5 Experimental results

The experiments and evaluations are conducted in the same way as previous studies using the E^{2}FGVI model [9]. The experimental data used is 50 DAVIS video object segmentation dataset and 50 randomly masked frames for video damage [13]. Also the global reference frames extracted uniformly at sampling rate of 10, and 5 preceding and succeeding local neighboring frames are used as inputs of the model. For quantitative metrics, we use LPIPS (Learned Perceptual Image Patch Similarity) and FID (Frechet Inception Distance) that evaluate the visual quality of the video from a human perspective [20, 4]. Additionally, we also use E$_{warp}$ (flow warping error) to measure temporal consistency [6]. LPIPS uses a pre-trained VGG or AlexNet to extract the features of the original video and the video that is being inpainted and compare their differences. We use AlexNet for feature extraction. And FID calculates the distribution of features from the original video and the inpainted video and measures the Wasserstein distance between the two distributions. The mean and covariance of each distribution are calculated by a pre-trained Inception-V3 model.

Experimental results, as shown in Table 1, reveal that decreasing the number of Transformer blocks from 8 to 4 results in a degradation in overall performance, but increases the model speed by approximately 40%. And when applying the proposed method to the model using 4 blocks, LPIPS and FID improve by approximately 2% and 2.6%, respectively. E_{warp} remains similar when using 4 blocks. This shows that the proposed method is an efficient way to improve performance while maintaining the model speed. To demonstrate the validity of the proposed method, we measure LPIPS and FID per 100K iterations as shown in Table 2, 3. It is observed that LPIPS and FID are improved at all measured iterations, which indicates that the proposed method is effective in improving the video inpainting performance.

As shown in Figure 3, qualitative evaluations are also conducted. After applying the proposed method, it is observed that the shape of lines and patterns are improved, resulting in reduced distortion of the video and visually plausible results. This shows that the proposed method is effective in various damaged video scenes, and can be applied in various fields.

Table 1. Quantitative results of Video Inpainting

Dataset	DAVIS			
Model	LPIPS ↓	FID ↓	E_{warp} $\times 10^{-2}$ ↓	Runtime (s/frame)
8 Blocks	0.0402	11.907	0.1315	0.095
4 Blocks	0.0445	12.958	0.1340	0.068
Our	**0.0436**	**12.628**	**0.1336**	**0.068**

Table 2. LPIPS measurement per 100K iterations

Dataset	DAVIS				
	LPIPS ↓				
Model \ Iterations	100K	200K	300K	400K	500K
4 Blocks	0.0593	0.0516	0.0503	0.0484	0.0445
Our	**0.0560**	**0.0494**	**0.0481**	**0.0458**	**0.0436**

Table 3. FID measurement per 100K iterations

Dataset	DAVIS				
	FID ↓				
Model \ Iterations	100K	200K	300K	400K	500K
4 Blocks	17.143	15.137	14.525	14.114	12.958
Our	**16.465**	**14.619**	**13.967**	**13.028**	**12.628**

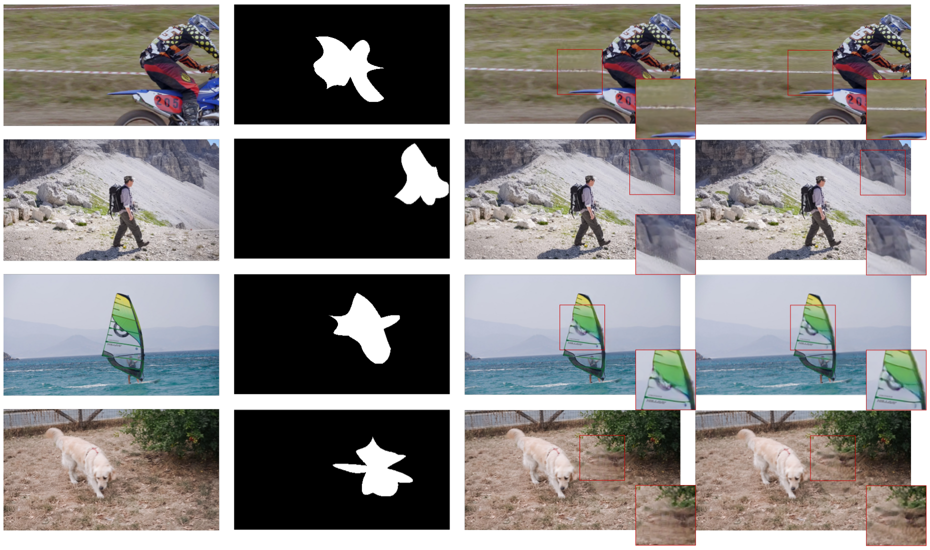

Fig. 3. Qualitative results of Video Inpainting. (First Column) Original Image, (Second Column) Mask for damaging the image, (Third Column) Inpainting results before using proposed method, (Forth Column) Video Inpainting results after using proposed method.

6 Conclusion

In this paper, we propose a method to improve video inpainting performance compared to the existing model E^{2}FGVI. The proposed method adds query to the Multi-head Self-Attention result to add local feature information. This method shows that it is an effective way to inpaint various natural images by improving the degraded performance as the number of Transformer blocks is reduced. However, the model speed is still not fast. If we further reduce the model size to increase the model speed, the model performance significantly degrades, leading to the need for a lightweight video inpainting model. In future research, we will focus on models that can operate in mobile environments while maintaining video inpainting performance.

References

1. Chang, Y.L., Liu, Z.Y., Lee, K.Y., Hsu, W.: Free-form video inpainting with 3d gated convolution and temporal patchgan. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9066–9075 (2019)
2. Chang, Y.L., Liu, Z.Y., Lee, K.Y., Hsu, W.: Learnable gated temporal shift module for deep video inpainting. arXiv preprint arXiv:1907.01131 (2019)
3. Gao, C., Saraf, A., Huang, J.B., Kopf, J.: Flow-edge guided video completion. In: European Conference on Computer Vision. pp. 713–729. Springer (2020)

4. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017)
5. Kim, D., Woo, S., Lee, J.Y., Kweon, I.S.: Deep video inpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5792–5801 (2019)
6. Lai, W.S., Huang, J.B., Wang, O., Shechtman, E., Yumer, E., Yang, M.H.: Learning blind video temporal consistency. In: Proceedings of the European conference on computer vision (ECCV). pp. 170–185 (2018)
7. Lee, S., Oh, S.W., Won, D., Kim, S.J.: Copy-and-paste networks for deep video inpainting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4413–4421 (2019)
8. Li, Y., Wu, C.Y., Fan, H., Mangalam, K., Xiong, B., Malik, J., Feichtenhofer, C.: Mvitv2: Improved multiscale vision transformers for classification and detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4804–4814 (2022)
9. Li, Z., Lu, C.Z., Qin, J., Guo, C.L., Cheng, M.M.: Towards an end-to-end framework for flow-guided video inpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17562–17571 (2022)
10. Liu, R., Deng, H., Huang, Y., Shi, X., Lu, L., Sun, W., Wang, X., Dai, J., Li, H.: Fuseformer: Fusing fine-grained information in transformers for video inpainting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14040–14049 (2021)
11. Oh, S.W., Lee, S., Lee, J.Y., Kim, S.J.: Onion-peel networks for deep video completion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4403–4412 (2019)
12. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2536–2544 (2016)
13. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 724–732 (2016)
14. Ranjan, A., Black, M.J.: Optical flow estimation using a spatial pyramid network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4161–4170 (2017)
15. Xu, N., Yang, L., Fan, Y., Yang, J., Yue, D., Liang, Y., Price, B., Cohen, S., Huang, T.: Youtube-vos: Sequence-to-sequence video object segmentation. In: Proceedings of the European conference on computer vision (ECCV). pp. 585–601 (2018)
16. Xu, R., Li, X., Zhou, B., Loy, C.C.: Deep flow-guided video inpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3723–3732 (2019)
17. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5505–5514 (2018)
18. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Free-form image inpainting with gated convolution. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4471–4480 (2019)
19. Zeng, Y., Fu, J., Chao, H.: Learning joint spatial-temporal transformations for video inpainting. In: European Conference on Computer Vision. pp. 528–543. Springer (2020)

20. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
21. Zhu, X., Hu, H., Lin, S., Dai, J.: Deformable convnets v2: More deformable, better results. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9308–9316 (2019)