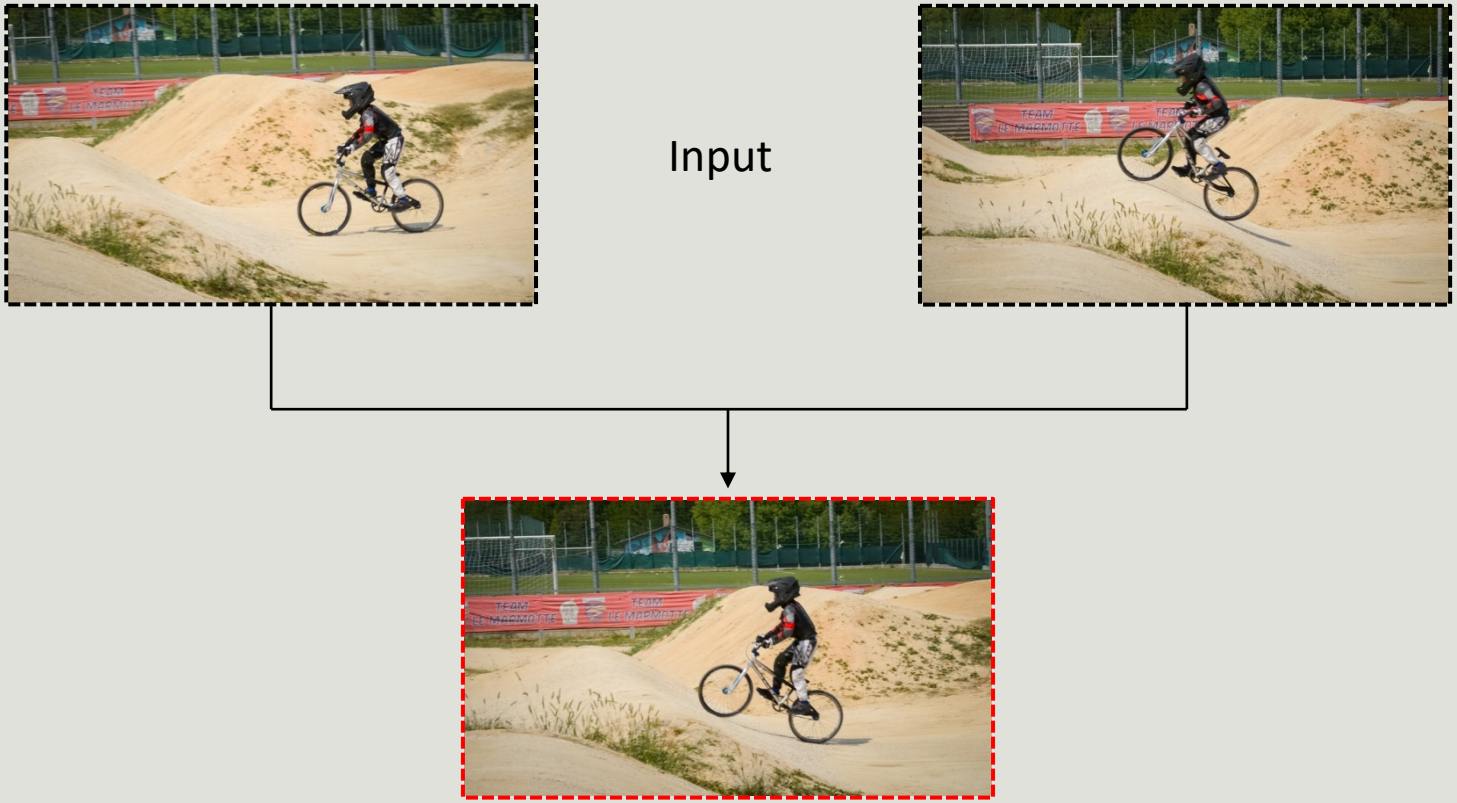
Residual Learning of Video Frame Interpolation Using Convolutional LSTM

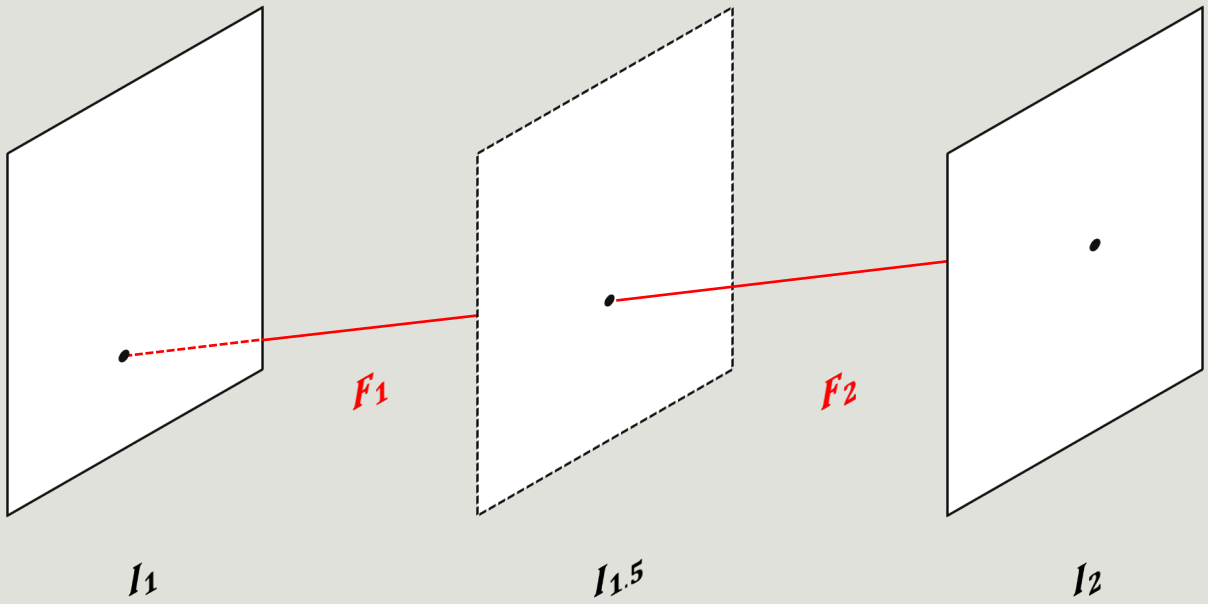
KEITO SUZUKI AND IKEHARA MASAAKI

KEIO UNIVERSITY (JAPAN)

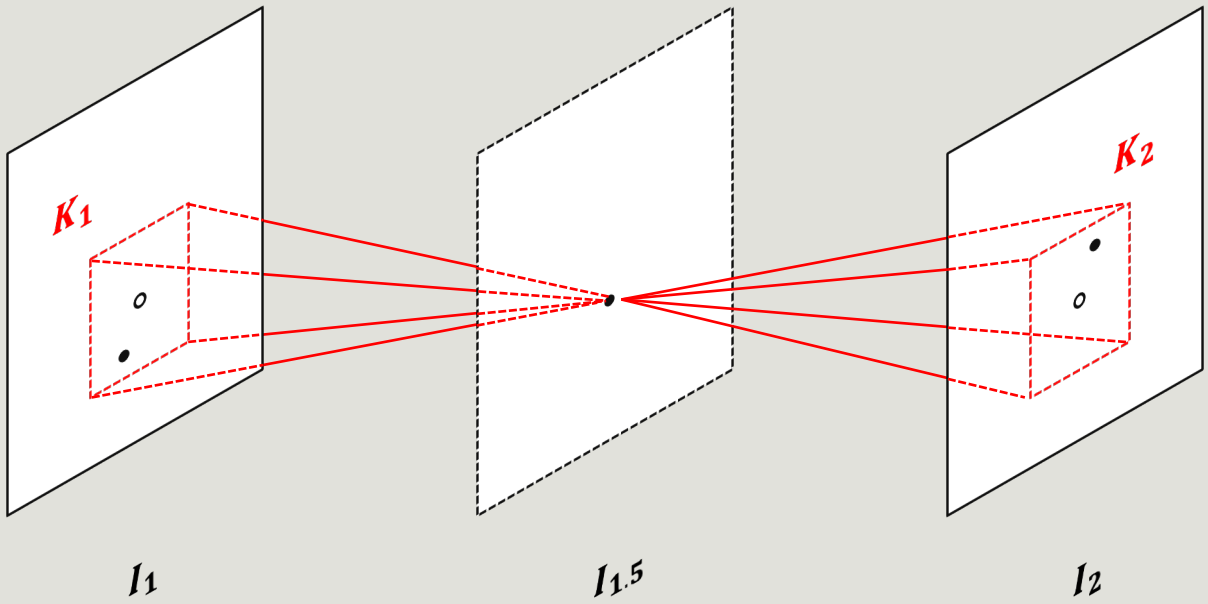
Video Frame Interpolation



Optical Flow Based Methods



Kernel Based Methods

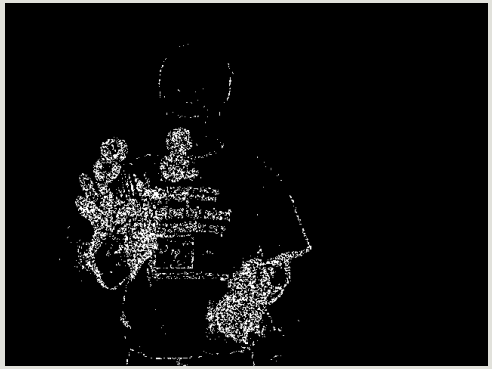


Proposed Method

- Residual learning
- Convolutional LSTM for feature extraction

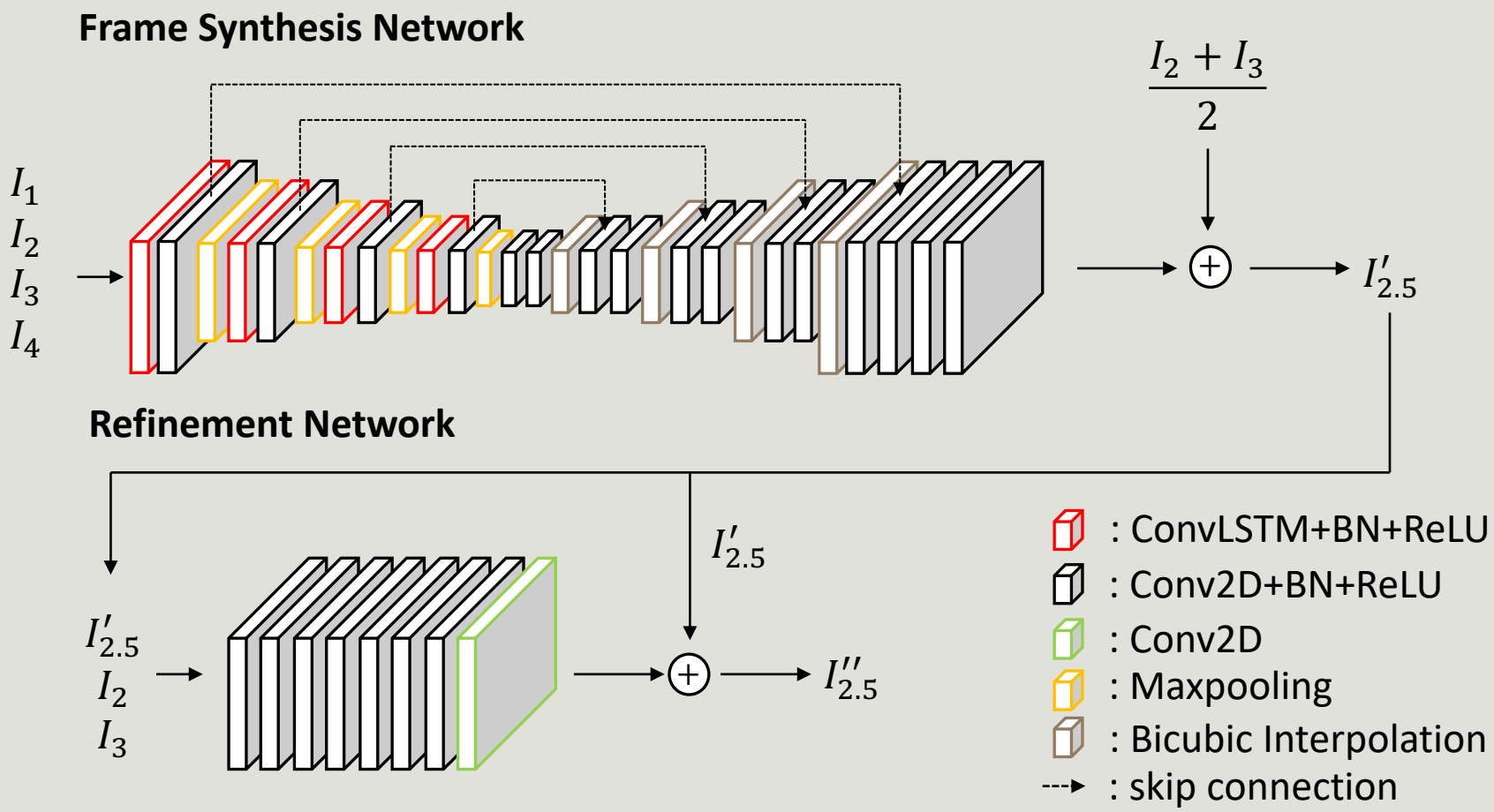
I_1

I_2



Difference

Network Structure



Implementation Details

Loss Function

$$L = 0.5 \cdot |I'_{2.5} - I_{2.5}| + | I''_{2.5} - I_{2.5}|$$

Dataset

- Vimeo90K Dataset (64620 7-frame sequences)
- Data augmentation: random crop (256×256), horizontal flip, reverse temporal order

Parameters:

- Learning Rate: 0.001
- $\beta_1 = 0.9,\ \beta_2 = 0.999$
- Epoch: 150

Quantitative Comparison

Middlebury Other Dataset

Video	SepConv [1]		CyclicGen [2]		DAIN [3]		Ours	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
DogDance	30.34	0.894	31.46	0.917	32.46	0.927	34.06	0.942
Minicooper	30.41	0.967	30.68	0.963	31.60	0.971	32.01	0.972
RubberWhale	45.25	0.992	42.43	0.985	44.09	0.990	44.55	0.989
Walking	31.02	0.940	33.58	0.953	34.25	0.960	36.54	0.965
Beanbags	29.76	0.926	28.19	0.913	32.84	0.937	31.43	0.933
Hydrangea	36.66	0.984	36.02	0.976	37.42	0.985	37.23	0.979
Grove2	36.45	0.979	32.20	0.921	36.85	0.980	36.28	0.975
Grove3	30.95	0.955	27.93	0.876	31.20	0.955	30.44	0.943
Urban2	41.10	0.984	29.58	0.829	43.62	0.989	34.19	0.924
Urban3	40.56	0.981	32.58	0.880	39.87	0.984	38.10	0.960
Average	35.25	0.960	32.45	0.921	36.42	0.968	35.48	0.958

Quantitative Comparison

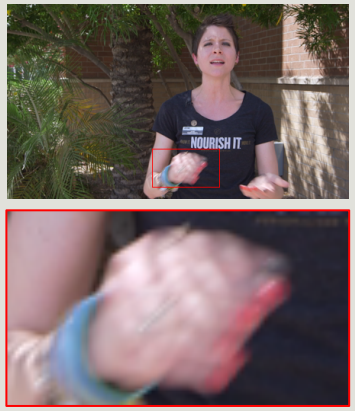
Vimeo90K and DAVIS 2017 Dataset

	Vimeo90K		DAVIS 2017	
	PSNR	SSIM	PSNR	SSIM
SepConv [1]	33.38	0.939	27.73	0.864
CyclicGen[2]	32.08	0.916	28.44	0.880
DAIN [3]	34.57	0.952	28.23	0.873
Ours	35.41	0.953	28.67	0.881

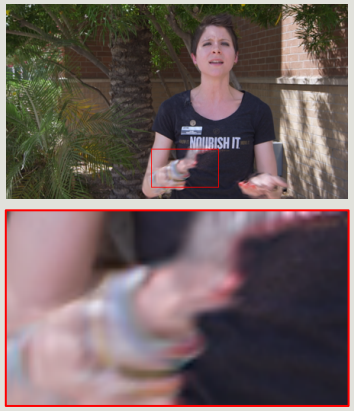
Qualitative Comparison

Input 1

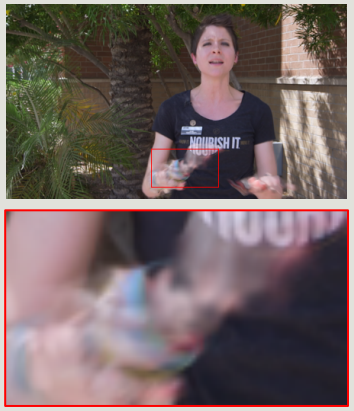
Input 2



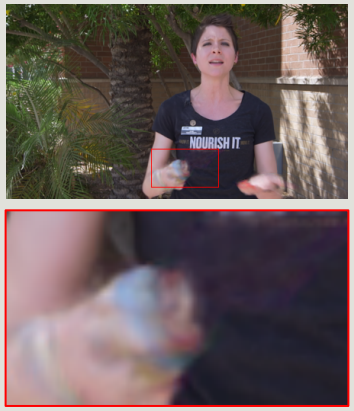
Ground Truth



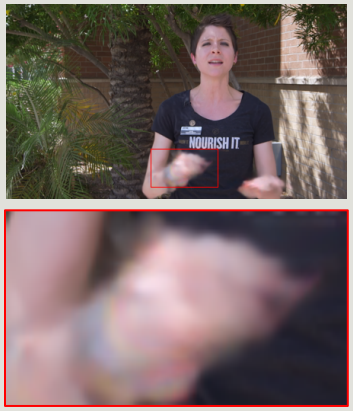
SepConv [1]



CyclicGen [2]



DAIN [3]

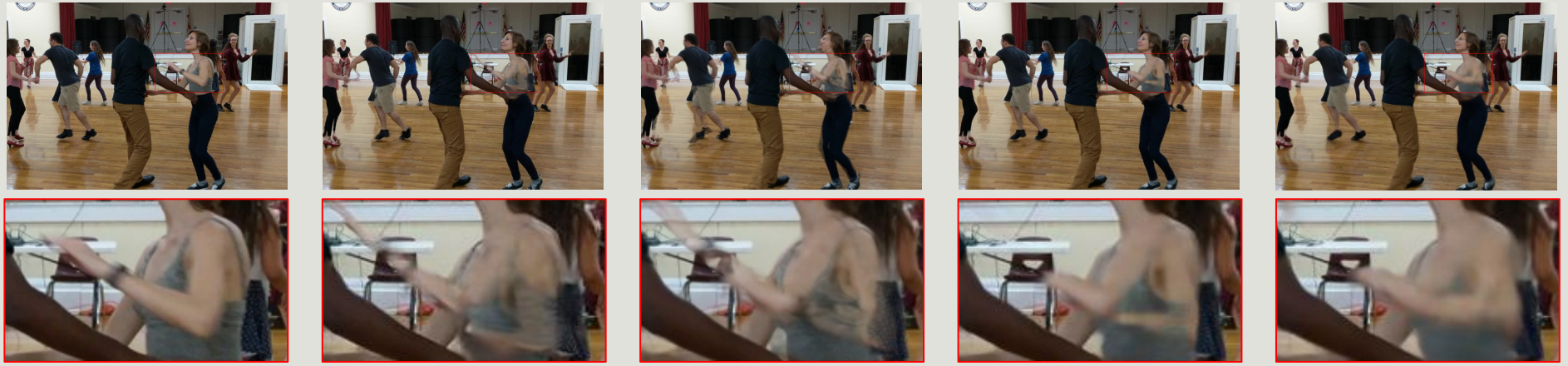


Ours

Qualitative Comparison

Input 1

Input 2



Ground Truth | SepConv [1] | CyclicGen [2] | DAIN [3] | Ours

Effectiveness of Our Model

Video	Baseline		+ConvLSTM		+Residual		+ConvLSTM +Residual		Complete Model	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
DogDance	29.25	0.869	29.91	0.884	33.26	0.940	33.32	0.939	33.41	0.938
Minicooper	25.72	0.874	26.52	0.892	30.69	0.963	30.73	0.963	31.52	0.969
RubberWhale	35.70	0.947	37.13	0.951	43.32	0.986	43.52	0.986	43.96	0.987
Walking	32.30	0.944	33.18	0.945	36.30	0.962	36.30	0.962	36.52	0.965
Beanbags	26.89	0.874	28.61	0.889	29.94	0.921	29.94	0.921	30.30	0.927
Hydrangea	32.16	0.888	32.92	0.918	35.40	0.951	36.12	0.953	36.71	0.971
Grove2	27.69	0.812	28.24	0.843	33.32	0.948	33.75	0.954	34.91	0.965
Grove3	25.24	0.792	26.15	0.835	28.61	0.908	28.71	0.909	29.40	0.923
Urban2	30.22	0.817	31.53	0.862	31.81	0.880	32.05	0.883	32.27	0.886
Urban3	30.70	0.824	32.93	0.883	35.37	0.924	35.22	0.922	36.02	0.934
Average	29.59	0.864	30.71	0.890	33.80	0.938	33.97	0.939	34.50	0.947

Conclusion

- ✓ Proposed a simple model that could compete against state-of-the-art methods for frame interpolation
- ✓ Effective for real world video sequences
- ✓ Generated frames still tend to be blurry

Reference List

[1] S. Niklaus, L. Mai, and F. Liu, “Video frame interpolation via adaptive separable convolution,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 261–270.

[2] Y.-L. Liu, Y.-T. Liao, Y.-Y. Lin, and Y.-Y. Chuang, “Deep video frame interpolation using cyclic frame generation,” in *AAAI Conference on Artificial Intelligence*, 2019.

[3] W. Bao, W.-S. Lai, C. Ma, X. Zhang, Z. Gao, and M.-H. Yang, “Depth-aware video frame interpolation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3703–3712.

Thank you for listening!

Contact: k_suzuki@tkhm.elec.keio.ac.jp