2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)

Self-Reproducing Video Frame Interpolation

Jiajun Deng[*1], Haichao Yu[*2], Zhangyang Wang[3], Xinchao Wang[4] Thomas Huang[2]

[1]*University of Science and Technology of China, Hefei, China*
[2]*University of Illinois at Urbana-Champaign, Urbana, USA*
[3]*Texas A&M University, College Station, USA*
[4]*Stevens Institute of Technology, Hoboken, USA*
dengjj@mail.ustc.edu.cn, haichao3@illinois.edu,
atlaswang@tamu.edu, xinchao.wang@stevens.edu, t-huang1@illinois.edu

***Abstract*—Frame interpolation has recently witnessed success by convolutional neural networks, that are learned from end to end to minimizing the reconstruction loss of dropped frames. This paper introduces a novel self-reproducing mechanism, that the real (given) frames could in turn be interpolated from the interpolated ones, to further substantially improve the consistency and performance of video frame interpolation. Such a consistency constraint accounts for the inherent symmetry between existing and interpolated frames in a video sequence, providing a strong form of self-supervision. We then build a pyramid-like architecture, under which existing interpolation models can plug-and-play as building blocks. Extensive experiments validate its state-of-the-art performance, on both high resolution videos in the wild and public benchmarks.**

***Keywords*-Frame Interpolation, Deep Learning, Self Supervision**

I. Introduction

Video data takes the majority of network flow among the multimedia applications. Frame interpolation, which renders information from temporally consecutive frames to construct one or more missing frames in between, has attracted increasing attention from the computer vision community [9], [35], [23], [27], [2], [26], [25]. Frame interpolation traditionally relied on optical flow or phase transition [9], [25], [37], [39]. Motivated by the success of convolutional neural networks (CNNs) on various image and video recognition problems [32], [18], [15], [33], [34], [36], [21], [20], [22], [38], [27], [26] adopt an end-to-end learned adaptive kernel convolving with input frame pairs to interpolate frames. At this point, with an end-to-end learning pipeline, we can grasp the relationship between input frame pairs and generate frames similar to real ones in color or phase space. However, due to lack of more effective constraint, such frameworks cannot guarantee the generated frames and real frames to be similar functionally and tends to be brittle to scenario changes such as fast motion and large object deformation, in which case the data distributions of the original and synthesized frames are left unsupervised and thus may differ significantly.

In this paper, we propose a novel mechanism for video frame interpolation called *Self-Reproducing Frame Interpolation* (SRFI). We impose the constraint that, during training, the interpolated frames must be able to reproduce the original frames that was used to produce those interpolated ones. This explicit self-reproduction constraint enforces the consistency across all video frames and represents the inherent symmetry between existing and interpolated frames. We show our illustration of SRFI in Figure. 1. In this figure, five frames from $Frame_t$ to $Frame_{t+4}$ are considered. We choose Adaptive Separable Convolution (Sepconv) [27] to show the effect of our method. To clarify, our algorithm can be exploited to improve upon any existing algorithm. Here, we use $Frame_t$, $Frame_{t+2}$ and $Frame_{t+4}$ as the inputs of the first level of our interpolation pyramid to get $\overline{Frame}_{t+1}$ and $\overline{Frame}_{t+3}$, both of which are then applied in next octave to synthesize $\overline{\overline{Frame}}_{t+2}$ using the same interpolation model.

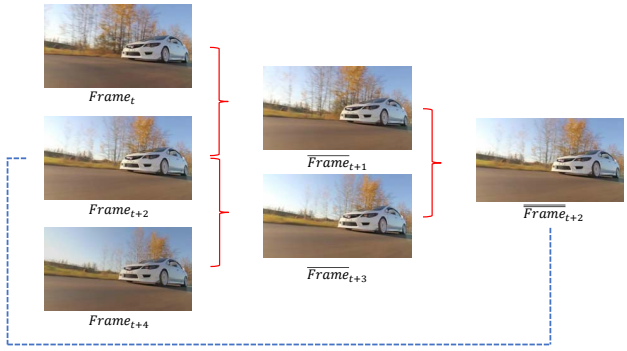

Figure 1. Illustration of our self-reproducing frame interpolation (SRFI). We enforce intermediate interpolated frames to infer back to the given real frames. In this example, we intend to make $\overline{\overline{Frame}}_{t+2}$ and $Frame_{t+2}$ as similar as possible.

In real video processing applications, interpolating just once may not meet the frame rate requirement. Examples include scalable video coding and transmission, where frame rates need to be dynamically adjusted at decoding, calling for the capability of iterative interpolation with few additional

978-1-7281-1198-8/19/$31.00 ©2019 IEEE
DOI 10.1109/MIPR.2019.00042

costs. Due to our self-reproducing nature that, our approach can be straightforwardly extended to iterative interpolation, with no extra training needed. To demonstrate our improvements in condition of iteratively producing multiple intermediate frames, we evaluate our results on Need For Speed (NFS) video benchmark [13] which features ultra-high frame rates (240fps) and thus the potential as a benchmark for multiple-time interpolation.

The similar idea of self-consistency has been recently exploited in several other tasks, including shape from stereo [12], tracking [11], [3], [1], [4]. Despite the existing work, we find it a novel and perfect fit for video frame interpolation. To our best knowledge, our approach is the first attempt to introduce the self-reproducing property to regularizing video interpolation models. Its power manifests in our impressive performance surpassing the state-of-the-art methods on *MiddleBurry* [5], NFS [13], and several other high-resolution videos. We believe the methodology to be of broad application value to the image/video restoration and enhancement fields.

Our contribution can be summarized into three-folds:

- We propose a novel self-reproducing video frame interpolation framework, that exploits the natural consistency prior between real frames and generated frames. The symmetric nature of video synthesis provides a strong form of self-supervision, that can be combined with and improved on state-of-the-art algorithms.
- We design a temporal pyramid-like architecture to fulfill the self-reproducing mechanism. A full set of training as well as data processing techniques has been developed accordingly.
- To our best knowledge, this is the first work to investigate iterative interpolation by reusing the same learned model, without needing any re-training. Our self-reproducing interpolation naturally fits this scenarios, and is shown to outperform state-of-the-art on iterative interpolation dramatically.

II. Related work

A. Frame Interpolation

Conventional frame interpolation consists of pixel-wise motion estimation and image synthesis [9], since it is easier to warp existing pixels from adjacent frames than to generate a new one from scratch [23]. Frame interpolation can be solved by warping pixels through flow field [6], [31], [5], [16] and the quality of interpolated frame is considered as an evaluation metric of optical flow estimation algorithm [7].

Frame interpolation algorithm based on optical flow fails in fast motion, occlusion and deformation cases due to failure of optical flow. Besides, phase-based approach [25] is proposed. However, the performance is also limited when large displacement occurs.

Recently, unsupervised optical flow estimation has been well studied [30], [16], and implicit motion estimation in frame interpolation are widely used. Deep Voxel Flow [23] implicitly estimates three dimensional voxel flow composed of two dimensional optical flow and another mask encoding amplitude information as an intermediate output in an end-to-end image synthesis processing. There is no supervision to directly guide voxel flow estimation, but using a per-pixel MAE loss with motion flow and mask regularizer to the final interpolation or extrapolation result. The intermediate voxel flow may not actually be the same as per-pixel motion information, but the end-to-end training fashion makes it converge to the state-of-the-art performance. In Sepconv [27], Niklaus *et al.* combine motion estimation and frame synthesis into one unified step. An adaptive spatial-varying kernel is learned for each output pixel to do frame interpolation end to end. The kernel can be decomposed into horizontal and vertical vectors, and a separable dynamic convolution operation is implemented so that each interpolated frame only takes one network forwarding.

Beyond delving to the pipeline or network structure for image synthesis, many supervision functions are designed [24], [35], [17]. For video prediction, Mathieu *et al.* propose a multi-scale prediction network and [24] tries to deal with edge blur in extrapolated images by adding image gradient difference and adversarial loss to general L_1 loss for sharper prediction. [35] combines Deep Voxel Flow [23] with multi-scale architecture and uses generative adversarial networks as supervision which get better interpolation results. Besides, Jason *et al.* [17] use VGG [33] trained on ImageNet [32] as feature reconstruction loss together with style reconstruction loss named perceptual loss to make better style transfer output. Perceptual loss is defined as L_2 distance between convolutional features between predicted and groundtruth images. Although adding perceptual loss may not improve PSNR, it usually improves visual quality.

B. Self-Supervision

Recently, fully supervised optimized neural networks have witnessed tremendous achievements in computer vision. However, it is labor-costly to collect large annotated datasets. Investigation of self-supervision comes as one path in the trends of saving labeling cost. Self-supervision is natural for video: for example, Zhu *et. al* [40] introduces event stream composed of grayscale images as supervising signal at training time with self supervision.

Self-supervision has not been adopted for low-level image and video processing until very lately. In a concurrent work to ours [8], Chen *et. al* extend traditional deblurring algorithms based on image formation and hand-crafted priors to utilize self-supervision information in blurring videos itself. An encoder-decoder architecture is designed to deblur the images first and then to be forced to use the estimated blur ones with clean images to regenerate the blur ones. The input of the encoder and the output of the decoder are supposed to be the same, which preserves consistency between the

degradation and inverse models. Our SRFI also investigates self-supervision but in a different flavor. Without any explicit motion estimation, we straightforwardly exploit the natural, simple symmetry between real and interpolated frames, so that the model learns to arrange motion consistently. We think video frame interpolation to be a more natural scenario for self-reproducing supervision, and are the first to explore here.

III. Self-Reproducing Frame Interpolation

The focus of our work is not on network architecture design. Instead, our main contribution is a training mechanism that can be adopted in different frame interpolation algorithm to achieve improvement. In this paper, we choose Sepconv [27] as our baseline to show that even with a strong enough framework, apparent improvement can be made using the self-reproducing supervision.

A. Intuition and Design

Generally, when we train a neural network for some task, we utilize groundtruth as supervision. In image synthesis task like frame interpolation, we use L_1 or L_2 distance and perceptual loss [17] to guide network optimization. L_1 loss leads to higher PSNR score while perceptual loss works better in terms of visual quality. Beyond these two choices, with the popularity of Generative Adversarial Networks [14], [10], [28], adversarial loss is considered to be a nice complement [24]. In this paper, we investigate self-reproducing supervision mechanism in frame interpolation. In the following paragraphs, we will explain our intuition and design.

To evaluate synthesized images, metrics including PSNR, SSIM and user ranking are widely used. However, these metrics cannot differentiate between functions of synthesized frames and original frames. Intuitively, a perfectly interpolated frame is one that can replace the real one in all aspects, which means we are not satisfied if the frames only look like real ones, we would like to use them like real ones. In [19], dehazed images are fed into Faster R-CNN [29] to detect objects where joint optimization proves to be effective. Classification and bounding box regression provide semantic supervision for dehazing and the end-to-end training leads to better performance. The most straightforward criterion to validate the effectiveness of interpolated frames is whether they can be used to generate inputs again. That is why we term our method as self-reproducing frame interpolation.

Our training scheme is composed of three steps as illustrated in Figure 2. In the first step, F_0 and F_2 are fed into Sepconv to synthesize $\overline{F_1}$, which is supervised by $Loss_1$ with the groundtruth F_1. At the same time, (F_2, F_4) and F_3 form another training triplet under the supervision of $Loss_3$. In the third step, we use interpolated frames $\overline{F_1}$ and $\overline{F_3}$ to further synthesize $\overline{\overline{F_2}}$ under the guidance of F_2. Note that F_2 functions as both input and output during training.

Loss function for each interpolated frame $\overline{F}_i$ is defined as a linear combination of L_1 distance in color space and perceptual loss in high-level feature space as below:

$$L_i = \lambda_1 \|F_i - \overline{F}_i\|_1 + \lambda_p L_p(F_i, \overline{F}_i). \quad (1)$$

Perceptual loss is L_2 distance between high-level feature maps defined as where

$$L_p(F_i, \overline{F}_i) = \|\delta(F_i) - \delta(\overline{F}_i)\|_2. \quad (2)$$

In our experiments, we use as δ the output of $conv4\_3$ in VGG16 [33]. Our overall training loss is

$$L = L_{t+1} + L_{t+3} + \alpha L_{t+2}. \quad (3)$$

In the above definitions, λ_1, λ_p, α are hyperparameters determining how each loss item contributes to the final objective function.

During model exploration, we considered only using $Loss_2$ and formulate this problem as a self-supervision setting. However, self-supervision alone leads to solution degradation. That is because the network simply learns an identity mapping to copy F_2 from input to output. To resolve this issue, we choose a self-reproducing fashion consisting of both self-supervision loss and intermediate reconstruction loss. In our experiments, we search the hyperparameter space for a relatively good set and present details in next section.

In our approach, self-reproducing training not only helps motion information preservation, but it also works well for iterative interpolation. During training, we use interpolated results to further synthesize intermediate frames again. Because of this, our model can be naturally applied to recursive frame interpolation. We will give a detailed comparison of iterative interpolation between our approach and other methods in the experiments.

In summary, we propose a self-reproducing frame interpolation method that utilizes interpolated frames to synthesize the original frames that are used to synthesize these interpolated ones. This recurrent interpolation mechanism leads to more accurate motion information preservation and is capable of recursive interpolation. Our algorithm is effective and simple to implement. As shown in next section, we achieve state-of-the-art performance on the evaluated datasets.

B. Data collection

To get high-resolution videos, we collect our training samples from YouTube. Similar to [27], we collect videos with several criterion: (1) We only use 1080P videos to guarantee high quality. (2) Frames on shot boundaries are not collected because these frames cannot be reconstructed by frame interpolation due to information loss. (3) Motion vector between frames should be large enough to to make the dataset challenging enough. After obtaining these frame samples, they are randomly sampled to 150×150 patches.

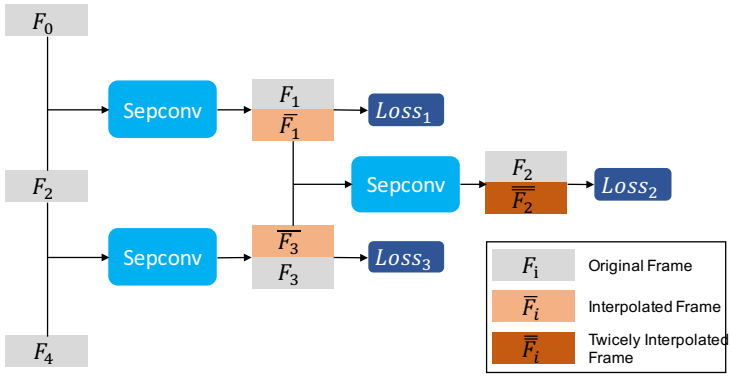

Figure 2. Self-reproducing training scheme. For simplicity, $Frame$ are represented by F. $F_{t,t+2,t+4}$ are real frames, while $\overline{F}_{t+1,t+3}$ are frames interpolated once and $\overline{\overline{F}}_{t+2}$ twice. We investigate self-reproducing training method with more frames containing several concurrent frames which are utilized as both inputs and the ground truth of final outputs. The exhaustive description of self-reproducing training is in Sec. III-A.

During training, these patches are further randomly cropped into 128×128 patches. In this way of online data augmentation, the network can be trained with less bias.

IV. Experiments

In this section, we evaluate our SRFI mechanism on several datasets and selected 1080P videos in the wild. Our approach yields results superior to the prior ones including the state-of-the-art methods.

A. Experiment Setting

We initialize our networks by Gaussian initialization with $\mu = 1, \sigma = 0.01$ and train with Adamax with $\beta_1 = 0.9$, $\beta_2 = 0.999$. As stated in Section III-A, three-step training strategy is employed. First, a Sepconv is trained with 20 epochs. After that, SRFI scheme is applied for 5 epochs with $lr = 1e^{-4}$. Finally, the entire end-to-end model is fine-tuned with $lr = 1e^{-5}$. Our training process is taken on a Nvidia GTX 1080Ti with 16 samples per batch.

Although we explicitly choose samples with large motion, data augmentation still contributes. Spatially, we randomly flip patches and shift patches to enhance motion vector's amplitude. Temporally, we reverse patches in time axis such that the order of input frames does not affect interpolated results.

B. Quantitative Results

Middleburry. We evaluate our method on Middlebury optical flow benchmark [5]. There are eight scenes in Middlebury with four conditions. Our approach surpasses other submissions on high-speed camera shot frames by a large margin. The evaluation results of high-speed camera part is shown in Table I, the metric in this dataset is Average Interpolation Error. Due to the self-consistency constrain of our training mechanism, the motion information and details are preserved better in our model.

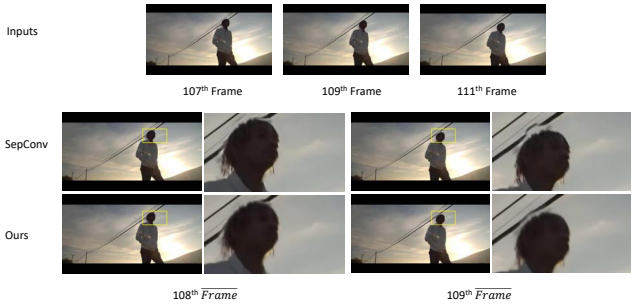

Figure 3. Visual results on *See You Again*. (Zoom out for more details.)

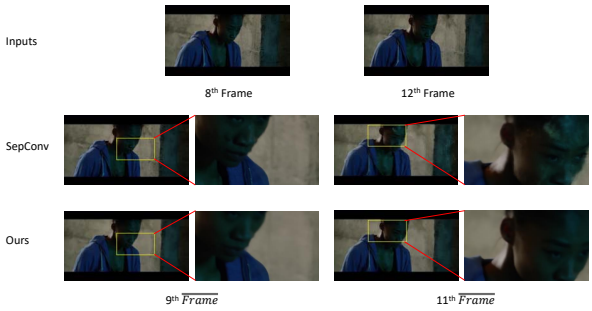

Figure 4. Visual results for iterative interpolation. (Zoom out for more details.)

YouTube Videos. Sepconv [27] is evaluated on the MV *See You Again*[1] to show its generalization ability to videos in the wild. In our experiments, we add two more MV videos

[1] https://www.youtube.com/watch?v=RgKAFK5djSk

from YouTube: *Shape of You*[2] and *Uptown funk*[3]. Watched more than 2.9B times, these are among the hottest videos on YouTube. The interpolation results are presented in Table II. We use PSNR to evaluate the performance for quantitative analysis. The results demonstrate that, for most cases, our SRFI outperforms Sepconv in real videos.

Recursive Interpolation. Given two input frames named as I_1 and I_9, we first interpolate the intermediate frame I_{1i9}. In the second interpolation iteration, two intermediate frames are interpolated as I_{1i1i9} and I_{1i9i9} between the three frames. For the third interpolation iteration, four intermediate frames are generated. They are named as $I_{1i1i1i9}$, $I_{1i1i9i1i9}$, $I_{1i9i1i9i9}$ and $I_{1i9i9i9}$. Since our self-reproducing model is trained with data augmentation, it is invariant to input frame order. The interpolation types are grouped in four categories: I^1: $\{I_{1i9}\}$, I^2: $\{I_{1i1i9}, I_{1i9i9}\}$, I^3: $\{I_{1i1i1i9}, I_{1i9i9i9}\}$ and I^4: $\{I_{1i1i9i1i9}, I_{1i9i1i9i9}\}$. The quantitative comparison is shown in Table III.

We evaluate our model for recursive interpolation on NFS dataset[13]. NFS dataset is a video dataset for high-frame rate object detection. We recursively interpolate 30fps videos three times and evaluate interpolation results using 240fps videos. In this setting, SRFI outperforms Sepconv with a large margin. It indicates the effectiveness of self-reproducing mechanism for iterative interpolation.

C. Qualitative Analysis of SRFI

In Figure. 3, we show the visual quality comparison of single-step interpolation on *See You Again*. We choose 107th and 109th frame to synthesize 108th frame, 109th and 111th frames to synthesize 110th frame. It is obvious that adding the self-reproducing mechanism helps avoid unnatural distortion of rigid objects.

Recursive interpolation visual comparison is shown in Figure 4. In this case, we use 8th and 12th frames from *Shape of You* to interpolate twice to synthesize 9th and 11th frames. As shown in the figure, when applied twice, Sepconv cannot properly handle illumination and textual reconstruction while our SRFI can work appropriately.

V. Conclusions

This paper proposes a self-reproducing mechanism for frame interpolation. Our essential idea is that, given a set of input frames and an interpolation model, the synthesized frames should, in turn, be able to produce the input frames by using that same model. Our mechanism explicitly enforces the symmetric constraint and thus provides strong self-supervision, making it especially effective for iterative interpolation. Furthermore, we build a pyramid-like architecture, allowing the existing interpolation algorithms to plug-and-play. Our method yields results superior to the state of the art on several standard benchmarks and wild videos from websites, in particular when recursive interpolation is needed.

[2]https://www.youtube.com/watch?v=JGwWNGJdvx8

[3]https://www.youtube.com/watch?v=OPf0YbXqDm0

References

[1] X. Wang, B. Fan, S. Chang, Z. Wang, X. Liu, D. Tao, and T. Huang. Greedy Batch-based Minimum-cost Flows for Tracking Multiple Objects. *IEEE Transactions on Image Processing* (TIP), 26:4765–4776, 2017.

[2] X. Wang, Z. Li, and D. Tao. Subspaces indexing model on Grassmann manifold for image search. *IEEE Transactions on Image Processing* (TIP), 20:2627–2635, 2011.

[3] X. Wang, E. Turetken, F. Fleuret, and P. Fua. Tracking Interacting Objects Optimally Using Integer Programming. In *European Conference on Computer Vision* (ECCV), September 2014.

[4] X. Wang, E. Turetken, F. Fleuret, and P. Fua. Tracking Interacting Objects Using Intertwined Flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (TPAMI), 38:2312–2326, 2016.

[5] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011.

[6] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, 2004.

[7] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *TPAMI*, 2011.

[8] H. Chen, J. Gu, O. Gallo, M.-Y. Liu, A. Veeraraghavan, and J. Kautz. Reblur2deblur: Deblurring videos via self-supervised learning. *arXiv:1801.05117*, 2018.

[9] B.-D. Choi, J.-W. Han, C.-S. Kim, and S.-J. Ko. Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation. *TCSVT*, 2007.

[10] E. L. Denton, S. Chintala, R. Fergus, et al. Deep generative image models using aw laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*, 2015.

[11] K. Fragkiadaki, W. Zhang, G. Zhang, and J. Shi. Two-granularity tracking: Mediating trajectory and detection graphs for tracking under occlusions. In *ECCV*, 2012.

[12] P. Fua. A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision and Applications*, 1993.

[13] H. K. Galoogahi, A. Fagg, C. Huang, D. Ramanan, and S. Lucey. Need for speed: A benchmark for higher frame rate object tracking. In *ICCV*, 2017.

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, 2014.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[16] J. Y. Jason, A. W. Harley, and K. G. Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In *ECCV*, 2016.

[17] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.

Table I
PART OF EVALUATION ON MIDDLEBURY BENCHMARK

	Backyard			Basketball			Dumptruck			Evergreen		
	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
SuperSlomo	**9.56**	**11.9**	3.30	5.37	10.2	2.24	6.69	15.0	**1.53**	6.73	10.4	1.66
Sepconv-v1	10.2	12.8	3.37	5.47	10.4	2.21	6.88	15.6	1.72	6.63	**10.3**	1.62
DeepFlow	11.0	13.9	3.63	5.91	11.3	2.29	7.14	16.3	1.49	7.80	12.2	1.70
SuperFlow	10.2	12.7	3.68	6.13	11.8	2.24	7.68	17.5	1.77	7.44	11.6	1.69
SRFI (ours)	10.3	12.9	**3.20**	**5.16**	**9.76**	**2.17**	**6.55**	**14.8**	1.65	**6.59**	**10.3**	**1.58**

Table II
ONE-TIME INTERPOLATION PSNR.

	See You Again	Shape of You	Uptown Funk
SRFI (0, 1, 0) (Sepconv L_f)	42.65	37.73	30.25
SRFI (0, 1, 0.1)	42.50	37.74	30.15
SRFI (1, 0, 0) (Sepconv L_1)	44.36	37.97	30.25
SRFI (1, 0, 0.1)	44.45	38.25	30.58
SRFI (1, 0.1, 0) (Sepconv)	44.38	37.96	30.58
SRFI (1, 0.1, 0.1)	44.00	37.98	30.43
SRFI (1, 1, 0) (Sepconv)	43.17	37.74	30.47
SRFI (1, 1, 0.1)	43.29	37.89	30.49

Table III
RECURSIVE INTERPOLATION PERFORMANCE ON NFS.

	I^1	I^2	I^3	I^4
Sepconv L_1	31.69	30.09	31.67	28.12
SRFI ($\lambda_1 = 1, \lambda_p = 0, \alpha = 0.1$)	33.13	33.12	33.75	31.24

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 2012.

[19] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng. An all-in-one network for dehazing and beyond. *arXiv preprint arXiv:1707.06543*, 2017.

[20] D. Liu, B. Cheng, Z. Wang, H. Zhang, and T. S. Huang. Enhance visual recognition under adverse conditions via deep networks. *arXiv preprint arXiv:1712.07732*, 2017.

[21] D. Liu, Z. Wang, Y. Fan, X. Liu, Z. Wang, S. Chang, and T. Huang. Robust video super-resolution with learned temporal dynamics. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2526–2534. IEEE, 2017.

[22] D. Liu, Z. Wang, Y. Fan, X. Liu, Z. Wang, S. Chang, X. Wang, and T. S. Huang. Learning temporal dynamics for video super-resolution: A deep learning approach. *IEEE Transactions on Image Processing*, 27(7):3432–3445, 2018.

[23] Z. Liu, R. Yeh, X. Tang, Y. Liu, and A. Agarwala. Video frame synthesis using deep voxel flow. In *ICCV*, 2017.

[24] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv:1511.05440*, 2015.

[25] S. Meyer, O. Wang, H. Zimmer, M. Grosse, and A. Sorkine-Hornung. Phase-based frame interpolation for video. In *CVPR*, 2015.

[26] S. Niklaus, L. Mai, and F. Liu. Video frame interpolation via adaptive convolution. In *CVPR*, 2017.

[27] S. Niklaus, L. Mai, and F. Liu. Video frame interpolation via adaptive separable convolution. In *ICCV*, 2017.

[28] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv:1511.06434*, 2015.

[29] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[30] Z. Ren, J. Yan, B. Ni, B. Liu, X. Yang, and H. Zha. Unsupervised deep learning for optical flow estimation. In *AAAI*, 2017.

[31] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *CVPR*, 2015.

[32] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015.

[33] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014.

[34] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, et al. Going deeper with convolutions. In *CVPR*, 2015.

[35] J. van Amersfoort, W. Shi, A. Acosta, F. Massa, J. Totz, Z. Wang, and J. Caballero. Frame interpolation with multi-scale deep loss functions and generative adversarial networks. *arXiv:1711.06045*, 2017.

[36] Z. Wang, S. Chang, Y. Yang, D. Liu, and T. S. Huang. Studying very low resolution recognition using deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4792–4800, 2016.

[37] Z. Wang, H. Li, Q. Ling, and W. Li. Robust temporal-spatial decomposition and its applications in video processing. *IEEE Transactions on Circuits and Systems for Video Technology*, 3(23):387–400, 2013.

[38] Z. Wu, Z. Wang, Z. Wang, and H. Jin. Towards privacy-preserving visual recognition via adversarial training: A pilot study. *arXiv preprint arXiv:1807.08379*, 2018.

[39] Z. Yu, H. Li, Z. Wang, Z. Hu, and C. W. Chen. Multi-level video frame interpolation: Exploiting the interaction among different levels. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(7):1235–1248, 2013.

[40] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis. Ev-flownet: Self-supervised optical flow estimation for event-based cameras. *arXiv:1802.06898*, 2018.