

FLAVR: Flow-Agnostic Video Representations for Fast Frame Interpolation

Tarun Kalluri *
UCSD

Deepak Pathak
CMU

Manmohan Chandraker
UCSD

Du Tran
Facebook AI

<https://tarun005.github.io/FLAVR/>

Abstract

Most modern frame interpolation approaches rely on explicit bidirectional optical flows between adjacent frames, thus are sensitive to the accuracy of underlying flow estimation in handling occlusions while additionally introducing computational bottlenecks unsuitable for efficient deployment. In this work, we propose a flow-free approach that is completely end-to-end trainable for multi-frame video interpolation. Our method, FLAVR, is designed to reason about non-linear motion trajectories and complex occlusions implicitly from unlabeled videos and greatly simplifies the process of training, testing and deploying frame interpolation models. Furthermore, FLAVR delivers up to 6x speed up compared to the current state-of-the-art methods for multi-frame interpolation while consistently demonstrating superior qualitative and quantitative results compared with prior methods on popular benchmarks including Vimeo-90K, Adobe-240FPS, and GoPro. Finally, we show that frame interpolation is a competitive self-supervised pre-training task for videos via demonstrating various novel applications of FLAVR including action recognition, optical flow estimation, motion magnification, and video object tracking. Code and trained models are provided in the supplementary material.

1. Introduction

Video frame interpolation [2, 9, 23, 27, 33, 37, 42, 43, 45, 78] aims to generate non-existent intermediate frames in a video between existing ones that are spatially and temporally coherent with the rest of the video, finding applications in overcoming the limited acquisition frame rate and exposure time of commercial video cameras. Traditionally, frame interpolation has been treated as a predominantly graphics problem where the approaches are complicated and hard coded. A large body of prior works use *flow warping* for frame interpolation [23, 43, 78], where the input frames are used to estimate (often bidirectional) optical flow maps from a pretrained flow prediction network, possibly along with

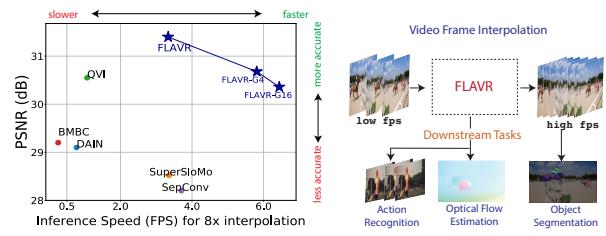


Figure 1. **Our contributions** We propose FLAVR, a simple and efficient architecture for single shot multi-frame interpolation. The plot of accuracy (PSNR) vs. inference speed (fps) of FLAVR compared with current methods on GoPro 8x interpolation with 512×512 input images. FLAVR is 6x faster than the current most accurate method (QVI) and 2x faster than the current fastest method (SuperSloMo) while maintains the same quality. FLAVR is also a useful self-supervised pretext task for various downstream applications.

additional information like monocular depth maps [2] and occlusion masks [3]. The frames at intermediate time steps are then interpolated either by using backward [2, 23] or forward warping [42, 43]. However, these optical flow-based approaches, as well as proposed alternatives [7, 27, 44, 45, 51], have to confront one or more of the following limitations:
1. **Computational Costs:** As they rely on optical flow and pixel level warping procedures, they are inefficient at both training and inference in terms of speed and efficiency making them less suitable for end applications. For example, QVI [77], DAIN [2] and BMBC [49] take order of seconds to generate frames for $8 \times$ interpolation (Figure 1) while requiring users to deploy custom CUDA kernels that prohibit seamless deployment across edge devices.
2. **Modeling Complex Trajectories:** The modeling capacity is limited to account for only linear [2, 23] or quadratic [8, 77] motion trajectories, and extending these to account for more complex motions is non-trivial using existing approaches.
+ 3. **Representation Inflexibility:** By accepting pre-computed optical flows as inputs, current methods focus on learning only spatial warping and interpolation, thus the representations learned in the process are not transferable to tasks beyond frame interpolation.

In this work, we aim to achieve a good trade-off between visual quality and inference speed for video interpolation. We do so by proposing FLAVR (Flow-Agnostic Video Rep-

*Work done during TK's internship at Facebook AI.

resentation network), which jointly addresses the aforementioned limitations. FLAVR is a simple, scalable approach for frame interpolation that utilizes spatio-temporal convolutions for predicting intermediate frames of a video. Without demanding access to external flow or depth maps, FLAVR can make end-to-end multiple-frame predictions in a single forward pass. It implicitly handles complex motions and occlusions through learning from large scale video data, significantly improving ease of deployment and inference speed compared to prior approaches (Figure 1, Figure 3a), while achieving state-of-the art interpolation accuracy (Table 1, Table 2).

We also posit that models learned from raw videos should be able to simultaneously reason about intricate synergy between objects, motions and actions for accurate frame interpolation. This is because different actions and objects have different motion signatures, and it is essential to precisely capture these properties through the representations learned for accurate frame interpolation. We ground this argument in the context of self-supervised representation learning from videos [10, 17, 50, 67]. While popular pretext tasks like frame ordering [12, 28, 40, 69, 73], pixel/color tracking [65, 68] or contrastive learning [14–16] are tailored to suit specific downstream applications, we show that frame interpolation offers a more generic representation learning objective owing to its combined motion and semantic understanding. To this end, we show the utility of FLAVR pretraining to improve performance on a variety of downstream tasks like action recognition, optical flow estimation and video object segmentation. In summary:

- We propose FLAVR, a scalable, flow-free, efficient 3D CNN architecture for video frame interpolation. To the best of our knowledge, FLAVR is the first video frame interpolation approach that is **both** *optical flow-free* and able to make *single-shot multiple-frame predictions* (Section 3).
- FLAVR is quantitatively and qualitatively superior or comparable to current approaches on multiple standard benchmarks including Vimeo-90K, UCF101, DAVIS, Adobe, and GoPro while offering the best trade-off in terms of accuracy and inference speed for video interpolation (Section 5, Figure 1 and 4).
- We demonstrate that video representations self-supervisedly learned by FLAVR can be useful for various downstream tasks such as action recognition, optical flow estimation and video object segmentation (Section 6).

2. Related Work

Video Frame Interpolation Video frame interpolation is a classical computer vision problem [35] and recent methods take one of phase based [37, 38], kernel based [7, 33, 44, 45, 51, 55], or flow based approaches, of which flow-based meth-

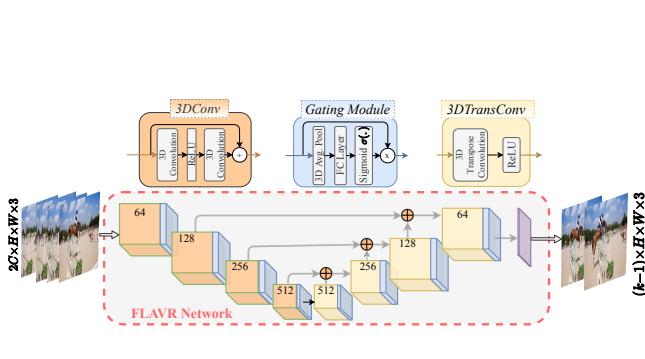
ods [2, 3, 8, 19, 23, 32, 42, 43, 57, 77–81] are most successful. The key idea in flow-based methods is to use a flow prediction network, *e.g.* PWC-Net [59], to compute bidirectional optical flow between the input frames [23] that guides frame synthesis along with predicting occlusion masks [3, 23, 78] or monocular depth maps [2] to reason about occlusions. While being largely successful in generating realistic intermediate frames, their performance is limited by the accuracy of the underlying flow estimator, which can be noisy in presence of complex occlusions resulting noticeable artifacts. They also assume uniform linear motion between the frames which is far from ideal for real world videos. Most importantly, the flow prediction and subsequent warping make frame prediction slow prohibiting fast interpolation. Recent works relax the linear motion assumption using quadratic warping [31, 77] at the cost of increased model complexity and inference time. CAIN [9] uses channel attention as suitable ingredient for frame interpolation but fails to capture complex spatio-temporal dependencies explicitly between input frames. Moreover, many recent methods are only aimed towards single frame interpolation [19, 57, 63]. We address all these issues in this work by designing an end to end architecture that directly predicts any number of intermediate frames from a given video by learning to reason motion trajectories and properties through 3D space-time convolutions while jointly optimizing for output quality and inference time.

Spatio-temporal Filtering Due to their proven success in capturing complex spatial and temporal dependencies, 3D space-time convolutions are very commonly used in video understanding tasks like action recognition [5, 11, 60, 62, 72], action detection [56, 74], and captioning [75]. We explore the use of 3D convolutions for the problem of temporal frame interpolation which requires modeling complex temporal abstractions between inputs for generating accurate and sharp predictions.

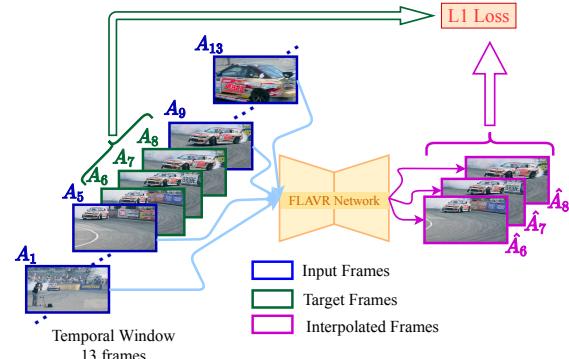
Video Self-Supervised Representation Learning Self-supervised learning deals with training unlabeled videos on artificial pretext tasks [10] to extract semantic representations that serve as useful priors for sparsely labeled downstream tasks. Videos contain rich source of information in the form of temporal consistency and frame ordering, and prior works make use of such cues to build pretext tasks like predicting ordering of frames [12, 28, 40, 69, 73], correspondence across time [22, 67] or contrastive predictive coding [15, 16]. In contrast to these approaches, we explore using video frame interpolation as a unified pretext task for both low-level and high-level downstream tasks like action recognition and optical flow.

3. Frame Interpolation using FLAVR

In video frame interpolation, the task is to generate a high frame-rate video from a lower frame-rate input video. We define k as the *interpolation factor*, where $k \times$ -video



(a) Overview of the proposed architecture



(b) Sampling procedure

Figure 2. FLAVR Architecture. (a) Our FLAVR is U-Net style architecture with 3D space-time convolutions (orange blocks) and deconvolutions (yellow blocks). We use channel gating after all (de-)convolution layers (blue blocks). The final prediction layer (the purple block) is implemented as a convolution layer to project the 3D feature maps into $(k-1)$ frame predictions. This design allows FLAVR to predict multiple frames in one inference forward pass. (b) A concrete example of our sampling procedure for $4\times$ interpolation ($k=4$) with 4-frame input ($C=2$). Best viewed in color.

frame interpolation corresponds to generating $(k-1)$ additional intermediate frames between every pair of original frames in the input video, that are both spatially and temporally consistent with the rest of the video. Prior approaches are either specifically designed for $2\times$ interpolation [9, 19, 27, 57, 63] or require multiple inferences for predicting all the k frames [2, 3, 49, 77]. In contrast, our aim is to design a framework which is simple yet enables single-shot $k\times$ -prediction for any value of k . Since training on, and generating, long videos are beyond the capacity of current hardware, we propose a simple sampling procedure for efficient training on raw videos, followed by the construction of the network architecture.

Sampling Training Data from Unlabeled Videos We can directly generate inputs and ground truths required for training from raw videos as follows. Let k be the interpolation factor, V is the original video with a frame rate f FPS. In order to generate training data for the $k\times$ -video frame interpolation problem, we sub-sample frames of V with a sampling stride of k to form a low frame rate video \bar{V} with $\frac{f}{k}$ fps. Then, to perform interpolation between any two frames at position $(i, i+1)$, given by A_i, A_{i+1} , of \bar{V} , we use a moving temporal window of size $2C$ in \bar{V} centered around A_i and A_{i+1} as the input, and all frames between A_i and A_{i+1} in original video V as the ground truth. This produces an input clip of size $2C$ frames (including A_i and A_{i+1}) and output clip of size $k-1$. FLAVR is flexible to handle any temporal context C instead of just the immediate neighbors A_i, A_{i+1} , which helps us to model complex trajectories and improve interpolation accuracy. The sampled input frames are concatenated in the temporal dimension resulting in input dimension $2C\times H\times W\times 3$, where H, W are the spatial dimensions of the input video.

An illustration of this sampling procedure is demonstrated in Figure 2b for the case of $4\times$ interpolation ($k=4$) with two context inputs from the past and future ($C=2$). In this

case, the frames $\{A_1, A_5, A_9, A_{13}\}$ are used as inputs to predict the 3 intermediate frames of $\{A_6, A_7, A_8\}$. Intuitively, the frames in the immediate neighborhood would be more relevant for frame interpolation than frames farther out. In our experiments, we find that for most common settings, using four context frames ($C=2$) is sufficient for accurate prediction on the datasets considered. We present a detailed study on the effect of the input context C in *supplementary material*.

Architecture Overview We present the proposed architecture of FLAVR in Figure 2a. FLAVR is a 3D U-Net obtained by extending the popular 2D Unet [53] used in pixel generation tasks, by replacing all the 2D convolutions in the encoder and decoder with 3D convolutions (3DConv) to accurately model the temporal dynamics between the input frames, invariably resulting in better interpolation quality. Each 3D filter is a 5-dimensional filter of size $c_i \times c_o \times t \times h \times w$, where t is the temporal size and (h, w) is the spatial size of the kernel. c_i and c_o are the number of input and output channels in the layer. The additional temporal dimension is useful in modeling the temporal abstractions like motion trajectories, actions or correspondences between frames in the video. We observed that our network indeed learns non-trivial representations along the temporal dimensions that can be reused in downstream tasks like action recognition with limited labeled data (Section 6).

Practically any 3D CNN architecture can be used as the encoder backbone, and we use ResNet-3D (R3D) with 18 layers [62] as our base backbone. We evaluate different variants of 3D CNNs with group convolutions [61] as backbones to achieve the best accuracy/speed trade-off and present the complete analysis and results in Figure 4. We remove the last classification layer from R3D-18, resulting in 5 conv blocks $conv1$ to $conv5$, each made up of two 3D convolutional layers and a skip connection. We also remove all temporal striding, as downsampling operations like striding and pooling

are known to remove details that are crucial for generating sharper images. However, we do use spatial stride of 2 in *conv1*, *conv3* and *conv4* blocks of the network to keep the computation manageable.

The decoder essentially constructs the output frames from a deep latent representation captured by the encoder by using progressive, multi-scale feature upsampling and feature fusion. For upsampling, we use 3D transpose convolution layers (*3DTransConv*) with a stride of 2. To handle the commonly observed checkerboard artefacts [47], we add a *3DConv* layer after the last *3DTransConv* layer. We also include skip connections that directly combine encoder features with the corresponding decoder along the channels to fuse the low level and high level information necessary for accurate and sharp interpolation.

The output of the decoder, which is a 3D feature map, is then passed through a temporal fusion layer, implemented by a 2D conv, in which the features from the temporal dimension are concatenated along the channels and fused into a 2D spatial feature map. This helps to aggregate and merge information present in multiple frames for prediction. Finally, this output is passed through a 7×7 2D convolution kernel that predicts output of size $H \times W \times 3(k-1)$, which is then split along the channel dimension to get the $(k-1)$ output frames. Our network is designed to efficiently handle interpolation for any value of k with minimum changes to the architecture.

Spatio-Temporal Feature Gating Feature gating technique is used as a form of self-attention mechanism in deep neural networks for action recognition [39, 72], image classification [18] and video interpolation [9]. We apply the gating module after every layer in our architecture. Given an intermediate feature dimension of size $f_i = C \times T \times H \times W$, the output f_o of the gating layer is given by $f_o = \sigma(W.pool(f_i) + b) \odot f_i$ where $W \in \mathbb{R}^{C \times C}$ and $b \in \mathbb{R}^C$ are learnable weight and bias parameters, *pool* is a spatio-temporal pooling layer and \odot is element-wise product along the channel dimension. Such a feature gating mechanism would suitably learn to upweight and focus on certain relevant dimensions of the feature maps that learn useful cues for frame interpolation, like motion boundaries.

Loss Function We can now train the whole network end to end using a pixel level loss like L1 loss between the predicted and ground truth frames, $\mathcal{L}(\{\hat{I}\}, \{I\}) = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^{k-1} \|\hat{I}_j^{(i)} - I_j^{(i)}\|_1$ where $\{\hat{I}_j^{(i)}\}$ and $\{I_j^{(i)}\}$ are the j -th predicted and the j -th ground truth frame of the i 'th training clip, k is the interpolation factor, and N is the size of the mini-batch used in training.

Representation Learning using FLAVR In order to successfully predict intermediate frames, it is essential for FLAVR to accurately reason about motion trajectories, estimate and capture motion patterns specific to objects, and reconstruct both high level semantic detail and low level

texture details. It is interesting to understand what types of motion information the networks learned and which tasks this representation is useful for. Therefore, we examine the possibility of using video frame interpolation in the context of unsupervised representation learning by pre-training FLAVR on the task of frame interpolation, and reusing the learned feature representations for the tasks of action recognition, optical flow estimation, and motion magnification. This objective serves the dual purpose of providing insights into the nature of representations learnt during training frame interpolation models, while also improving the performance of downstream tasks compared to random initialization.

4. Experimental Setup

Datasets. We use *septuplets* from the Vimeo-90K dataset [78] extracted from 30FPS videos for training single frame interpolation networks ($k=2$). We train our model on the train split and evaluate it on the test split of the dataset. Following [77], we additionally verify the *generalization* capability of our proposed approach. For single frame interpolation, we report the performance of a model trained on *Vimeo-90K* on the 100 quintuples generated from UCF101 [25] and 2,847 quintuples generated from DAVIS dataset [52]. For multi frame interpolation, we use Go-Pro [41] as the training set, and report results on the Adobe dataset [58] and GoPro dataset [41] for $8 \times$ interpolation.

Training Details. We use a R3D-18 backbone as the standard encoder in FLAVR. We also evaluate different variants of 3D CNNs with group conv [61] as backbones to achieve the best accuracy/speed trade-off. For data augmentation, we exploit the symmetry of the problem by randomly selecting input sequences during training and inverting the temporal order of the frames. Also, we also horizontally flip all frames of randomly selected inputs. Our hyper-parameter choices and more training details are provided in *supplementary*.

Evaluation Metrics. Following previous works, we use PSNR and SSIM metrics to report the quantitative results of our method. For multi-frame interpolation we report the average value of the metric over all the predicted frames, and also additionally report the TCC (Temporal Change Consistency) [8]. Since these quantitative measures do not strongly correlate with the human visual system [46], we also conduct a user study to analyze and compare our generated videos with other competing approaches.

Baselines. We perform comparisons with the following baselines that perform single and multi frame video interpolation. (i) **DAIN** [2] performs depth aware frame interpolation. (ii) **QVI** [77] computes quadratic flow prediction and adaptive filtering. (iii) **DVF** [33] uses volumetric sampling to generate the output frames. (iv) **SepConv** [45] predicts optimum pairs of spatially varying kernels for generating frames using input resampling. (v) **SuperSloMo** [23] performs warping based on flow and visibility maps. (vi) **CAIN** [9] performs

Method	Inputs	Vimeo-90K		UCF101		DAVIS	
		PSNR (\uparrow)	SSIM(\uparrow)	PSNR(\uparrow)	SSIM(\uparrow)	PSNR(\uparrow)	SSIM(\uparrow)
DAIN [2]	RGB+Depth+Flow	33.35	0.945	31.64	0.957	26.12	0.870
QVI [77]	RGB+Flow	35.15	<u>0.971</u>	<u>32.89</u>	<u>0.970</u>	<u>27.17</u>	0.874
DVF [33]	RGB	27.27	0.893	28.72	0.937	22.13	0.800
SepConv [45]	RGB	33.60	0.944	31.97	0.943	26.21	0.857
CAIN [9]	RGB	33.93	0.964	32.28	0.965	26.46	0.856
SuperSloMo [23]	RGB	32.90	0.957	32.33	0.960	25.65	0.857
BMBC [49]	RGB	34.76	0.965	32.61	0.955	26.42	<u>0.868</u>
AdaCoF [27]	RGB	<u>35.40</u>	<u>0.971</u>	32.71	0.969	26.49	0.866
FLAVR	RGB	36.25 \pm 0.06	0.975	33.31 \pm 0.02	0.971	27.43 \pm 0.02	0.874

Table 1. Comparison with state-of-the-art methods for 2x interpolation on Vimeo-90K, UCF101, and DAVIS datasets. The upper table includes the methods that use additional networks trained to predict optical flows and/or depth maps. The lower table represents the methods that use only RGB as input. The first and second best methods are marked in **bold** and underlined text. Our method consistently outperforms prior works which take only RGB as input, as well as works which additionally require optical flows and/or depth inputs.

Method	Inputs	Adobe		GoPro	
		PSNR	SSIM	PSNR	SSIM
DAIN [2]	RGB+Depth+Flow	29.50	0.910	29	0.91
QVI [77]	RGB+Flow	33.68	0.97	<u>30.55</u>	0.933
DVF [33]	RGB	28.23	0.896	21.94	0.776
SuperSloMo [23]	RGB	30.66	0.391	28.52	0.891
FLAVR	RGB	<u>32.20</u>	<u>0.957</u>	31.31	0.94

Table 2. Comparison with state-of-the-art methods for 8x interpolation on Adobe and GoPro datasets. FLAVR outperforms all previous work that use only RGB as input.

frame interpolation by channel attention and sub-pixel convolutions, (vii) **AdaCoF** [27] uses adaptive collaboration of flows, and (viii) **FLAVR** is our proposed approach. We could not compare against recent works like SoftSplat [43], AAO [8] and RRPN [82] as their training code is not available online for retraining on our setting.

Comparison across baseline models. We note that each of these prior works report their numbers using a different training and testing setup in their respective papers, so the numbers differ among various works. For example, DAIN [2] and AdaCoF [27] train and test on *triplet-split* of Vimeo-90K while SuperSloMo [23] and QVI [77] train their models on private custom datasets. To ensure fairness and a unified evaluation testbed, we accounted for all these variations by *retraining baseline models* for [2, 9, 23, 27, 33, 77] till convergence on *septuplet-split* of Vimeo for comparison in Table 1. Likewise, in Table 2, we retrained the presented baselines on GoPro data for fair comparison.

5. How does FLAVR compare with the state-of-the-art?

Single-Frame Interpolation. We report the results for single frame interpolation in Table 1, corresponding to 2x($k=2$) interpolation from 15 FPS to 30 FPS. We observe that FLAVR outperforms prior methods by a significant margin on Vimeo-90K dataset and sets the *new state-of-the-art*

on this dataset with a PSNR value of 36.25 and SSIM value of 0.975. FLAVR is a more generally applicable method and outperforms [23, 33, 45] which assume uniform linear motion between the frames. FLAVR also performs better than [9] which uses a similar end to end architecture to predict output frames, underlining the benefits achieved using an encoder-decoder architecture with spatio-temporal kernels. More importantly, FLAVR also outperforms DAIN [2] and QVI [77] without demanding additional knowledge in the form of bidirectional flow or depth maps.

We test the generalization capability of our method by evaluating the same trained model on UCF101 and DAVIS datasets. These are relatively more challenging for video frame interpolation, containing complex object and human motions from a range of dynamic scenes. Nevertheless, with a PSNR of 33.33 on the UCF101 dataset and 27.44 on the DAVIS dataset, FLAVR clearly delivers better performance compared to all the baselines methods which take RGB images as inputs, and performs on par or better than methods that additionally demand depth or flow maps as inputs. These datasets together constitute a wide spectrum of difficulty in terms of complex motions and occlusions, and FLAVR outperforms other methods on all the settings.

Multi-Frame Interpolation. For multi-frame setting, we report results on 8x ($k=8$) interpolation in Table 2, which corresponds to going from 30 to 240 FPS by generating 7 intermediate frames. Our method yields a PSNR of 31.31 and an SSIM score of 0.94 on the GoPro dataset, which is better than all the prior approaches proposed for frame interpolation. On the Adobe dataset, our method outperforms all methods significantly except QVI, but QVI additionally uses an optical flow estimator which helps on the more challenging Adobe dataset. Additionally, we evaluate TCC [8] on GoPro to obtain 0.78, 0.76, 0.73 for FLAVR, QVI, DAIN respectively. It is evident that FLAVR outperforms those prior works. AOO [8] reports 0.83, but it is trained on cus-

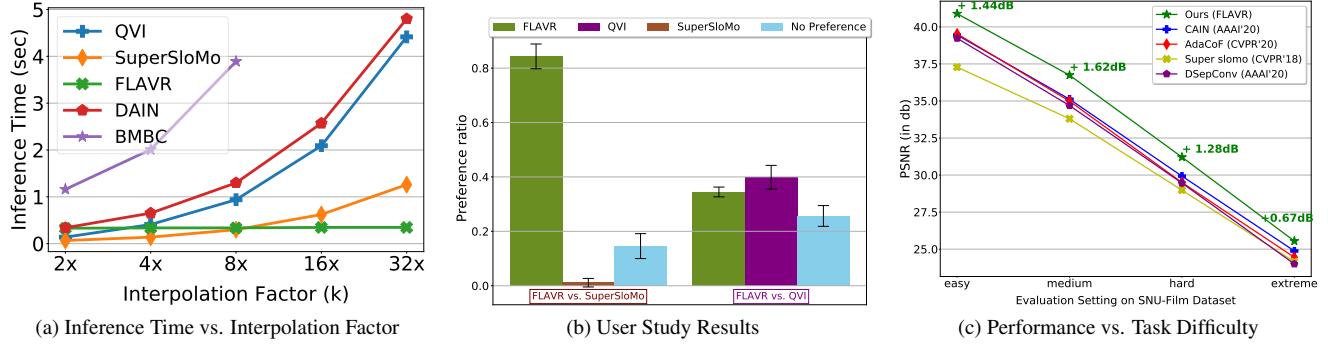


Figure 3. **Analysis.** (a) Inference time (forward pass w/o IO) of different methods on different interpolation factor. FLAVR has almost no change in inference time due to its design to predict multiple frames per inference. (b) Comparison between FLAVR with Super-SloMo and QVI in a user study on DAVIS. FLAVR significantly outperforms Super-SloMo, and performs comparable to QVI. (c) Comparison between FLAVR with other methods on SNU-Film dataset. FLAVR consistently outperforms all comparing methods across all levels of task difficulty.

tom data and uses GAN loss, which is biased in favor of this metric (and GAN loss is complementary to FLAVR and other VFI methods). Similar improvements in performance can also be observed in the case of $4\times$ ($k=4$) interpolation, as shown in the supplementary material. Additionally, we show qualitative results by using FLAVR on few sequences from DAVIS dataset in Figure 5. These results indicate the effectiveness of the proposed FLAVR architecture for the task of multi-frame interpolation.

Results on Middlebury We evaluate FLAVR on the publicly available test images from Middlebury [1, 54] dataset on the task of single frame interpolation. FLAVR is ranked **2nd, 5th, 8th** on *backyard, evergreen, basketball* sequences respectively, at the time of this submission. The complete results are available on the public leaderboard ([link](#)). Qualitative comparisons with other approaches on Middlebury images are provided with the *supplementary* material.

Speed vs. Accuracy Trade-off. One major challenge for realizing the applications of video frame interpolation for real time applications on low resource hardware is to optimize the trade off between faster inference speed and better interpolation quality. Perhaps the most important contribution of our work is to propose an approach that strikes an optimum balance between both these factors by achieving best performance with smallest runtime. As shown in Figure 1, FLAVR offers an improved run time for multi-frame interpolation models. This improvement is possible mainly because we require no overhead in terms of computing optical flow or depth, and predict all the frames in a single forward pass. We also show in Figure 3a that the inference speed using our method scales gracefully with an increase in the interpolation factor k , while most prior methods incur linear growth with k . We achieve runtime improvements of $2.7\times$, $6.2\times$ and $12.7\times$ for $8\times$, $16\times$ and $32\times$ interpolation respectively with respect to QVI, which is the current most accurate method, while providing much higher interpolation accuracy compared to SuperSlomo, which is the current

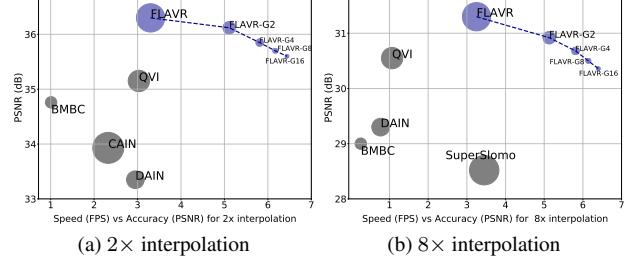


Figure 4. **Speed, accuracy and parameter tradeoff comparison.** Speed (in FPS, on x-axis) vs. accuracy (in PSNR, on y-axis) for various baselines as well as various architecture choices for FLAVR. Number of parameters in each model is proportional to the size of the marker. (a) is for $2\times$ and (b) is for $8\times$ interpolation. FLAVR-G x corresponds to FLAVR with x number of group convolutions. In summary, FLAVR achieves best speed-accuracy tradeoffs compared to many recent methods.

fastest.

We also perform an in-depth ablation on the effect of using group convolutions [61] on the speed-accuracy trade-offs on FLAVR, and showcase results in Figure 4. Specifically, for every 3D conv block, we replace the residual block by a channel separated convolution block [61] with groups $g = 1, 2, 4, 8$ and 16 , indicated by FLAVR, FLAVR- $2\times$, FLAVR- $4\times$ and so on in Figure 4. Note that $g = 1$ refers to our default setting in all other experiments. We show the results on Vimeo-90K for $2\times$ interpolation as well as GoPro dataset on $8\times$ interpolation. We find that compared to baselines that deliver similar performance (eg. QVI), *FLAVR is at least 6 \times faster on 8 \times interpolation* (refer Figure 4b, FLAVR-G8 vs. QVI). Furthermore, compared to baselines that give similar inference time speeds, *FLAVR delivers at least 3dB accuracy gain* (refer Figure 4b, FLAVR vs. SuperSlomo). These results indicate that FLAVR is a flexible architecture achieving best speed accuracy trade-off for video frame interpolation compared to existing methods.

Robustness to Task Difficulty. We validate the robustness in performance of our method using the SNU-Film dataset [9] consisting of videos with varying difficulty for in-

Model	PSNR	SSIM
R2D-18-2I	33.98	0.966
R2D-18-4I	34.97	0.967
R3D-18-4I	36.3	0.975

(a) Effect of encoder arch.

Model	PSNR	SSIM
No fusion	35.1	0.9713
fusion - add	35.7	0.9737
fusion - concat	36.3	0.975

(b) Type of feature fusion

Model	PSNR	SSIM
w/o stride	36.3	0.975
w/ 2x stride	35.4	0.961
w/ 4x stride	35.21	0.96

(c) Effect of temporal striding

Model	PSNR	SSIM
L1 Loss	36.3	0.975
L2 Loss	35.3	0.965
Huber Loss	35.3	0.964
L1+VGG Loss	35.91	0.962

(d) Effect of loss function

Table 3. **Ablation results** for FLAVR architecture on (a) different backbones, (b) fusion methods, (c) temporal striding, and (d) loss functions.

Figure 5. **Qualitative comparison with state-of-the-art methods.** We qualitatively compare FLAVR with Super-SloMo (SSM), QVI on a few video sequences on DAVIS. More qualitative results and generated videos are provided along with the supplementary material.

terpolation depending on the temporal gap between the input frames. The four settings we use are *easy* (120-240 FPS), e.g. predicting 240 FPS video from 120 FPS input, *medium* (60-120 FPS), *hard* (30-60 FPS) and *extreme* (15-30 FPS). In Figure 3c, we compare the performance of our method with prior works including CAIN [9] and AdaCoF [27], and report better performance than all the methods consistently across all the difficulty settings. Specifically, we see a gain of 1.28dB and 1.62dB compared to the next best approach [9] in the *hard* and *medium* settings respectively, which are considered challenging for video frame interpolation because of large motions and longer time gaps between the frames indicating robustness to video frame rates.

User Study. We conduct a user survey on Amazon Mechanical Turk to analyze the performance of our method in comparison to [77] and [23] on the 90HD videos from Davis dataset for 8× interpolation. We explain the details of the survey in *supplementary*, and summarize the results in Figure 3b. Firstly, when the comparison is between our method against Super-SloMo, users overwhelmingly preferred our videos as the generated videos looked more realistic with minimum artefacts around edges and motion boundaries owing to accurate interpolation. In comparison with QVI, users choose FLAVR in 35% of videos compared to QVI, which was chosen in 40% of the videos; and for 20% of videos the differences came out to be negligible. These results further support our hypothesis that in the interest of real

world deployment, optical flow and warping based frame interpolation methods can be substituted with our learning based approach that offers faster inference (Figure 3a) with minimal loss in performance.

Ablations. We provide detailed ablation into various design choices of the architecture, network and loss functions on the Vimeo-90K dataset in Table 5, and enlist the salient observations here. Firstly, we find that compared to an encoder with 2D Resnet-18 which takes a channel-wise concatenation of 4 images, FLAVR gives a 1.3dB gain on PSNR (Table 5a) validating our choice of spatio temporal network. Also, we find that using no striding in the temporal dimension (36.3dB) performs better than using stride of 2 (35.4dB) or 4 (35.21dB), supporting the hypothesis that temporal striding hurts in capturing sharp pixel level detail (Table 5c). Likewise, we observe that adding VGG-based perception loss [24] to the L1 losses during training is inferior in terms of PSNR (Table 5d). We include additional results on the effects of channel gating along with supporting qualitative results with the supplementary material.

6. How useful is FLAVR in enabling downstream applications?

Action Recognition We evaluate the semantic properties of the internal representations learned by FLAVR by reusing its trained encoder on a downstream action recognition task. We remove the decoder and attach a classification layer to the network. The whole network is then finetuned end to end on UCF101 and HMDB51 datasets. In order to isolate the benefits arising from pretraining the encoder on video interpolation task, we train a 3D resnet (R3D) baseline completely from scratch and observe from Table 4a that FLAVR, which is pretrained on Vimeo-90K dataset on frame interpolation task clearly outperforms random initialization baseline by 13.08% on UCF-101 and 4.48% on HMDB-51. FLAVR also significantly outperforms prior self-supervised methods on video which use low level pretext tasks like VideoGAN [64] and flow descriptors [34] indicating that frame interpolation can learn useful motion representations. Finally, FLAVR also achieves better accuracy than pretraining using DVF [33] which underlines that our particular method for video frame interpolation invariably benefits downstream action recognition more than voxel flow.

Optical Flow Estimation It is known that successful frame interpolation intrinsically depends on reliable opti-

Method	pretrained on	Arch.	UCF101	HMDB51
Random Init.	-	R3D-18	50.02	19.00
Supervised	Kinetics-400	R3D-18	87.70	59.10
Contrastive [15]	Kinetics-400	R3D-18	68.20	34.50
Video-GAN [64]	UCF101	Custom	52.10	-
LMD [34]	NTU RGB	Custom	53.00	-
DVF [33]	UCF101	Custom	52.40	-
FLAVR	Vimeo-90K	R3D-18	63.10	23.48

(a) Action recognition.

Table 4. **FLAVR for various downstream applications.** (a) FLAVR as a self-supervised pretext task for action recognition on UCF101 and HMDB51. (b) for optical flow prediction on MPI (Sintel [4]) and Kitti [36] datasets. (c) for video object segmentation mask propagation for low fps DAVIS videos. \mathcal{J}_m measures the region similarity as mean IoU, while \mathcal{F}_m is a boundary alignment metric.

Dataset	FlowNet [20]	Random Init.	Finetune on FLAVR
MPI-Clean [4]	2.02	4.41	2.92
MPI-Final [4]	3.14	5.27	3.90
Kitti-12 [13]	4.09	9.25	5.23
Kitti-15 [36]	10.06	17.22	13.68

(b) Optical flow estimation.

(c) Video object segmentation mask propagation.



Figure 6. **Video object segmentation mask propagation on DAVIS.** FLAVR helps to improve video object tracking in low fps videos. FLAVR is first used to up-sample video into higher frame rate, then a standard object segment propagation, e.g., CRW [21], is applied on interpolated videos. Refer Table 4c for quantitative improvements.

cal flow estimation [71]. We investigate this hypothesis by finetuning our trained network for optical flow estimation on MPI Sintel [4] and Kitti [13, 36] datasets, and report the corresponding EPE (end point error) in Table 4b. Finetuning using FLAVR achieves much lower EPE compared with random initialization using the same backbone architecture, proving that our model learns useful flow features. We note that we do not aim to outperform more complex, flow-dedicated architectures [20, 30] but aim to understand if we can learn useful flow features using a simple architecture like ours by pre-training on frame interpolation.

FLAVR improves VOS at low fps So far we evaluated FLAVR’s representation quality for downstream task but how good is its raw output in improving downstream applications? To study this, we consider the task of video object segmentation label propagation where the task is to propagate object masks throughout the video by extracting visual correspondences [21, 29, 68, 76]. Most of current approaches which perform label propagation assume access to 30FPS videos during training and testing (for example, from DAVIS), but the ability to find correspondences, and hence the accuracy of label propagation, falls considerably if the inputs are from low fps videos. We show that in such cases, FLAVR can be used to improve the accuracy of video object segmentation (VOS). To this end, we subsample the test videos from DAVIS dataset by $2 \times$ ($30\text{FPS} \rightarrow 15\text{FPS}$) and $4 \times$ ($30\text{FPS} \rightarrow 8\text{FPS}$) factors, and then apply the label propagation algorithm proposed in CRW [21]. Additionally, we also apply FLAVR for frame interpolation with $k = 2, 4$ to recover the original 30FPS videos in each case respectively, and apply the CRW algorithm again on the upsampled videos. From Table 4c and Figure 6, we observe that FLAVR can be effectively used as an intermediate step to improve the

results of label propagation on low fps videos. More details regarding the experiment are present in the *supplementary*.

Motion Magnification Motion magnification [48, 70] is a complementary problem to frame interpolation, in which the task is to magnify the subtle motions from the input video. Using FLAVR as pre-training, we fine-tune a motion magnification network for a fixed magnification factor of 10. FLAVR achieves an SSIM of 0.801 on the synthetic CoCo-Synth dataset [48] with a simple architecture. More details and qualitative results are presented in the *supplementary*.

7. Discussion

We present FLAVR for flow-free and end-to-end video frame interpolation. FLAVR uses 3D convolutions to model the spatio-temporal relations between the input frames improving interpolation accuracy under challenging motion profiles across various input frame rates. In extensive experiments and analysis presented across the main paper and the supplementary, we show that FLAVR offers best trade-off in terms of inference speed vs. interpolation accuracy compared to many existing approaches. We show that the representations learned by FLAVR are useful for various downstream tasks such as action recognition, optical flow estimation, video object segmentation mask propagation and motion magnification. We also invite the reviewers to look at more qualitative results and generated videos which are provided along with the **supplementary material**.

FLAVR still requires retraining for each interpolation factor k although for most of the practical applications, it is well known beforehand what would be the desired interpolation factor. Being a data-driven end-to-end approach, FLAVR shares with other deep learning based approaches the limited generalization capability to data outside the training distri-

bution. Nevertheless, we expect FLAVR to stimulate new directions for frame interpolation with ample opportunity for simpler and efficient methods to address these limitations.

8. Ablations

In Table 5, we present a detailed ablation study of the proposed architecture design in terms of the skip connections, strides and loss functions. In addition to the brief insight provided in the main text, we explain each of them in detail next. We conduct all the ablation studies on the Vimeo-90K dataset.

Backbone Architecture In this work, we propose using 3D convolutions that model space-time relations for improved frame interpolation. To verify this hypothesis, we train a video interpolation network using 2D convolutions instead, and present the results in Table 5a. While training 2D Resnet, we concatenate RGB channels of the input before feeding into the network. We observe that the *R2D-18-2I* baseline, which uses a 2D ResNet-18 encoder decoder with 2 input frames ($C = 1$) performs worse than *2D-R18-4I* baseline, which uses 4 input frames ($C = 2$) justifying the need for a larger input context. Next, our proposed architecture *3D-R18-4I* which uses 3D convolutions along with 4 inputs, clearly outperforms both these baselines by 1.3 and 2.3dB, respectively. This indicates the importance of temporal modeling for the task.

In Figure 7, we present a more detailed ablation about the effect of input context (C) on the performance of interpolation. From Figure 7, we observe that for both $2\times$ and $8\times$ interpolations, using two input frames ($C = 1$), one each from past and present is sub-optimal, as it fails to accurately reason about complex motion profiles and occlusions. Furthermore, for $2\times$ interpolation, we found that a value of $C = 2$ gave the best result, and beyond that the performance saturates. This is because the outer frame generally contain less useful information for interpolation and in some cases might contain significant scene shifts which hurts the interpolation accuracy. In the case of $8\times$ interpolation, the time gap between the frames is tinier, so we find that a value of $C = 3$ performs the best, while any larger value of C hurts the accuracy.

Choice of Fusion Table 5b compares and reports the different choices for the skip connection (in Figure 2 of the main submission) used for combining features across encoder and corresponding decoder. *No fusion* corresponds to having no skip connection between the layers of the encoder and decoder. While *fusion - add* corresponds to adding the features from the encoder to the decoder, *fusion - concat* refers to concatenating the corresponding feature maps along the channels. We find that using some kind of feature transfer across encoder and decoder is essential, than having *No fusion* (PSNR of 36.11 vs. 35.1), as the complementary information learnt in the low level and high level features

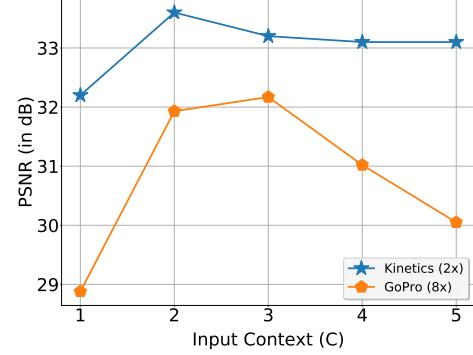


Figure 7. Effect of Input Context Comparison of the effect of input context, C , for video frame interpolation. For $2\times$ interpolation, we observed that a value of $C = 2$ which corresponds to using 4 input frames, 2 each from the past and future, gives best results. Beyond $C = 3$, we observe no further improvements. For $8\times$ interpolation, a value of $C = 3$ gave the best accuracy.

needs to be aggregated for accurate interpolation. We settle on using *fusion - concat* in our final model as it gives better performance than *fusion - add*.

Temporal Striding Striding or pooling in CNNs are known to remove lot of fine level details in images, which are essential for generative tasks like frame interpolation. We verify this with experiments using $2\times(1/2\times)$ and $4\times(1/4\times)$ temporal striding in the encoder(decoder), and observe from Table 5c that the performance decreases from 36.3 to 35.2 with larger temporal striding. We conclude that temporal striding hurts, and use a temporal stride of 1 in all the 3D convolution layers.

Channel Gating We visualize the role of channel gating module in the network in Figure 8. We show the overlapped input frames in Figure 8a to highlight the parts which have motion. In Figure 8b and Figure 8c, we plot the feature maps corresponding to the channel dimension with the largest activation while using and without using the feature gating respectively. We observe that the network trained with spatio-temporal gating (Figure 8b) learns to focus on parts of input with visible motions (high activations in red), thus resulting in confident predictions of the interpolated motion estimates compared to Figure 8c. In fact, training without spatio-temporal gating results in a drop in PSNR value from 36.3 to 36.1, further validating the utility of having the gating module.

Loss Function Many previous works [45] have studied the effect of using purely pixel loss vs. perception based losses [24]. Using only L1 or L2 loss would improve on the PSNR metric, but would cause blur in predictions. On the other hand, adding VGG based perception loss would result in sharper images visually. We observe from Table 5d that we did not improve upon the PSNR or SSIM metric by using any additional loss functions like VGG loss or Huber

Model	PSNR	SSIM
R2D-18-2I	33.98	0.966
R2D-18-4I	34.97	0.967
R3D-18-4I	36.3	0.975

(a) Effect of encoder arch.

Model	PSNR	SSIM
No fusion	35.1	0.9713
fusion - add	35.7	0.9737
fusion - concat	36.3	0.975

(b) Type of feature fusion

Model	PSNR	SSIM
w/o stride	36.3	0.975
w/ 2x stride	35.4	0.961
w/ 4x stride	35.21	0.96

(c) Effect of temporal striding

Model	PSNR	SSIM
L1 Loss	36.3	0.975
L2 Loss	35.3	0.965
Huber Loss	35.3	0.964
L1+VGG Loss	35.91	0.962

(d) Effect of loss function

Table 5. **Ablation results** for FLAVR architecture on (a) different backbones, (b) fusion methods, (c) temporal striding, and (d) loss functions.Figure 8. **Visualization of attention weighted feature maps** (a) The overlayed input frames. (b) The feature map of the channel with the highest attention weight in the network with feature gating. (c) The same feature map without using the gating module. We observe higher activation (red) in (b) along the motion boundaries. Best viewed in color.

loss, apart from just L1 loss which also resulted in visually sharper images in our case.

9. Experiment Settings for downstream applications

9.1. Low-fps video object segmentation details

To examine the effectiveness of using the outputs of FLAVR, we choose the task of object segmentation in videos using mask propagation.

Motivation Achieving good label (or mask) propagation requires estimating perfect pixel level correspondences between frames of a video, using similarity between the respective feature maps. However, estimating such correspondences might be challenging if the frame sequences are extracted from low-fps videos. We want to validate if using FLAVR can improve low-fps video object segmentation.

Setup and baseline. DAVIS is the standard benchmark popularly used for video object segmentation which include videos at 30FPS. To adopt to low-fps setup, we purposely downsample DAVIS videos into lower frame rates, e.g., 15FPS or 8 FPS, and evaluate different object segmentation approaches on these low-fps videos. We choose CRW [21], the current state-of-the-art method for video object segmentation, as a baseline which is applied directly on downsampled low-fps videos. We then compare this baseline with using the same method, i.e. CRW, on interpolated videos generated by FLAVR by 2x or 4x interpolation from 15FPS or 8FPS videos. Results are shown in the main submission

showing that FLAVR helps to improve low-fps video object tracking. The label propagation mechanism is the same as used in [21].

9.2. Motion magnification

Motion Magnification [48, 66, 70] deals with magnifying subtle yet important motions from videos, which are often imperceptible by human eyes. From [48], we define motion magnification as follows. For an Image $I(\mathbf{x}, t) = f(\mathbf{x} + \delta(\mathbf{x}, t))$, the goal of motion magnification is to generate an output image $\tilde{I}(\mathbf{x}, t)$ such that

$$\tilde{I}(\mathbf{x}, t) = f(\mathbf{x} + (1 + \alpha)\delta(\mathbf{x}, t)) \quad (1)$$

for a magnification factor α . For frame interpolation, $\alpha < 1$, since we are interested in what happens between two frames while for motion magnification, $\alpha > 1$, since we look to extrapolate existing motions beyond visible regime. While prior works [48, 66, 70] used custom architectures along with various post processing filters for this task, we offer a complementary perspective and look into how much a simple architecture like FLAVR pretrained on frame interpolation helps motion magnification. For this purpose, we use the synthetic dataset *CoCo-Synth* [48] to perform the training. We train the network for a fixed magnification factor of 10 ($\alpha = 10$). On this dataset, when compared to no pretraining at all, pretraining on FLAVR improved the SSIM values on a held-out validation set from 0.732 to 0.801. We provide sample videos after magnification and compare it with phase

based approach [70] in our supplementary video. We emphasize that we do not apply any post processing such as temporal or spatial filters for removing noise on the outputs. The videos are generated directly as an output of the FLAVR architecture pretrained on frame interpolation, and finetuned for motion magnification.

9.3. Experiment setting for action recognition

For downstream experiments on action recognition, we use the train and validation split 1 of UCF101 [25] and HMDB51 [26]. We remove the decoder from the architecture and use the pretrained encoder along with a classifier (a global average pooling, a fully-connected layer, and a softmax) for training on downstream actions and add a temporal stride of 4. For UCF101, we use an input size of $32 \times 3 \times 224 \times 224$ and for HMDB51 we use an input size of $16 \times 3 \times 224 \times 224$ with a batch size of 16. The networks are fine-tuned using SGD with batch norm with a learning rate of 0.02 for 40 epochs. During inference, we sample 10 consecutive overlapping clips of length 32 from the test video and average predictions over all the clips.

9.4. Experiment setting for optical flow estimation

One crucial point to consider in downstream training on optical flow is that the flow networks generally take only two input frames which is considered too short for 3D CNN. Nevertheless, to examine the effectiveness of features learnt using frame interpolation for optical flow, we use the same encoder and decoder, and initialize the last prediction layer to output two channels instead (corresponding to x and y values of flow at each pixel). Since the interpolation network was trained to take 4-frame inputs, we apply copy padding to the inputs, e.g. repeating each input frame 2 times. We use an EPE (end point error) loss and train our network for 200 epochs. We report numbers using 5-fold cross validation over the MPI-Sintel clean and fina as well as kitti subsets.

10. Qualitative Results

We show additional qualitative results by applying frame interpolation technique on insect motion videos in Figure 9. We believe that this application is of immense use for closer inspection of biological properties from videos. We obtain videos from AntLab Youtube channel¹ that have insect take-off and flying captured at very high FPS. We down-sample the frame rate to 15FPS and apply our interpolation network to recover videos of higher frame rate. We apply our $8\times$ model once to obtain videos of 120FPS. The images are shown in Figure 9. Complete videos are available in our supplementary video.

Middlebury Dataset.

¹<https://www.youtube.com/user/adrianalansmith>

We evaluate FLAVR on the publicly available test images from Middlebury dataset [54] on the task of single frame interpolation. However, Middlebury has test samples with only two input frames while FLAVR requires 4-frame inputs. In those examples, we simply duplicate them into 4 frames and evaluate with FLAVR. For two frame sequences like *teddy*, duplicating inputs is obviously sub-optimal. On sequences where multi-frame inputs are available, FLAVR outperforms most prior interpolation works like SuperSloMo [23], BMBC [49] and EDSC [6]. Qualitative results for some such sequences are presented in Figure 10. The complete results are available on the [public leaderboard](#).

11. User study

We carry the user study on the Amazon Mechanical Turk (AMT) platform. We select two representative works that belong to two broad families that perform linear (SuperSloMo [23]) and quadratic (QVI [77]) warping for multi-frame interpolation. Then, we compare each video generated by FLAVR with videos generated using each of SuperSloMo and QVI separately. For this purpose we use *all* 90 HD videos from the DAVIS dataset, generate $8\times$ interpolated videos and place the two interpolated videos one beside the other and randomly shuffle the order of videos. We then show each pair of videos to 6 AMT workers and ask them to choose which video, right or left, looked more realistic. The method preferred by more users is chosen as a winner for that particular video. In case of tie, that is if each method is chosen by 3 users, we place the video under “no preference” category. Workers are paid in accordance with minimum wages rules. With this setting, we find that users overwhelmingly chose our videos in preference against SuperSloMo [23]. More details are provided in subsection 5.1 of the main paper.

12. Training details

We train the $2\times$ interpolation network on Vimeo-90K dataset and $4\times$ and $8\times$ interpolation networks on the GoPro dataset and use the official train and validation splits with the sampling strategy explained in subsection 3 of the paper. We use a crop size of 256×256 and 512×512 for Vimeo-90K and GoPro datasets, respectively. We employ random frame order reversal and random horizontal flipping as augmentation strategies on both the datasets. We use an initial learning rate of 2×10^{-4} and divide the learning rate by 2 whenever the training plateaus. We train the $2\times$ interpolation network for 200 epochs, while $4\times$ and $8\times$ interpolation network were trained for 120 epochs. We use a mini-batch size of 64 on Vimeo-90K dataset and 32 on GoPro dataset, and train our network on 8 2080Ti GPUs. We reduce the learning rate by half whenever the training

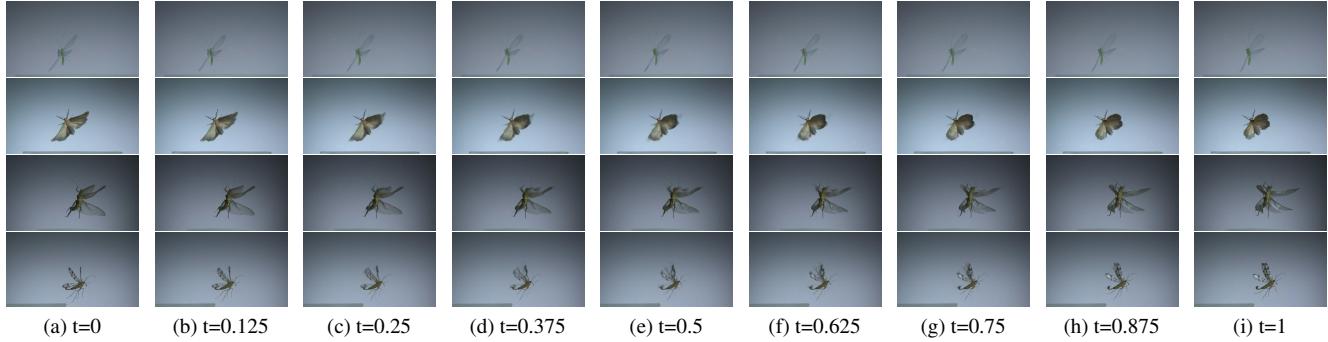


Figure 9. Qualitative Results for $8\times$ video frame interpolation on Insect Motion Videos. Frame at $t = 0$ and $t = 1$ are given as inputs to the network to predict the remaining 7 intermediate frames. Original Videos acquired from AntLab Youtube Channel.

plateaus which is cross-validated by the validation set. We apply mean normalization once for every mini-batch of input frames separately rather than using global mean normalization or batch normalization inside the network to achieve training stability. We use 8 GPUs and a mini-batch of 32 to train each model, and training is completed in about 36 hours for $2\times$ and 22 hours for $8\times$ interpolation networks.

13. Benchmarking inference time

The inference time benchmarking was performed using an NVIDIA-2080Ti GPU with 12GB memory. The calculated time only includes forward pass excluding the data pre-processing time and CPU/GPU transfer. The results were obtained by averaging over 100 samples from Adobe-240FPS dataset using 512×512 crop size. For multi-frame interpolation, the time required is calculated as the aggregate time required for interpolating all the frames. Non-blocking CUDA operations as well as GPU warm start time were accounted for during inference time computation.

14. Statement on potential negative impact

Frame interpolation aims to generate non-existent frames between existing frames of a video. While achieving state-of-the-art performance using simple architectures through FLAVR is a plus, any kind of generative models can be misused to forge or tamper a video which may have a negative impact on applications where outputs of FLAVR have a bearing on reliability. Moreover, one of the applications of FLAVR is to improve object tracking in videos, which might have a potential to be used in surveillance for nefarious purposes.

References

- [1] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International journal of computer vision*, 92(1):1–31, 2011. 6
- [2] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3703–3712, 2019. 1, 2, 3, 4, 5
- [3] Wenbo Bao, Wei-Sheng Lai, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Memc-net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 1, 2, 3
- [4] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.), editor, *European Conf. on Computer Vision (ECCV)*, Part IV, LNCS 7577, pages 611–625. Springer-Verlag, Oct. 2012. 8
- [5] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 2
- [6] Xianhang Cheng and Zhenzhong Chen. Multiple video frame interpolation via enhanced deformable separable convolution. *arXiv preprint arXiv:2006.08070*, 2020. 11, 13
- [7] Xianhang Cheng and Zhenzhong Chen. Video frame interpolation via deformable separable convolution. In *AAAI*, pages 10607–10614, 2020. 1, 2
- [8] Zhixiang Chi, Rasoul Mohammadi Nasiri, Zheng , Juwei Lu, Jin Tang, and Konstantinos N Plataniotis. All at once: Temporally adaptive multi-frame interpolation with advanced motion modeling. *arXiv preprint arXiv:2007.11762*, 2020. 1, 2, 4, 5
- [9] Myungsub Choi, Heewon Kim, Bohyung Han, Ning Xu, and Kyoung Mu Lee. Channel attention is all you need for video frame interpolation. In *AAAI*, pages 10663–10671, 2020. 1, 2, 3, 4, 5, 6, 7

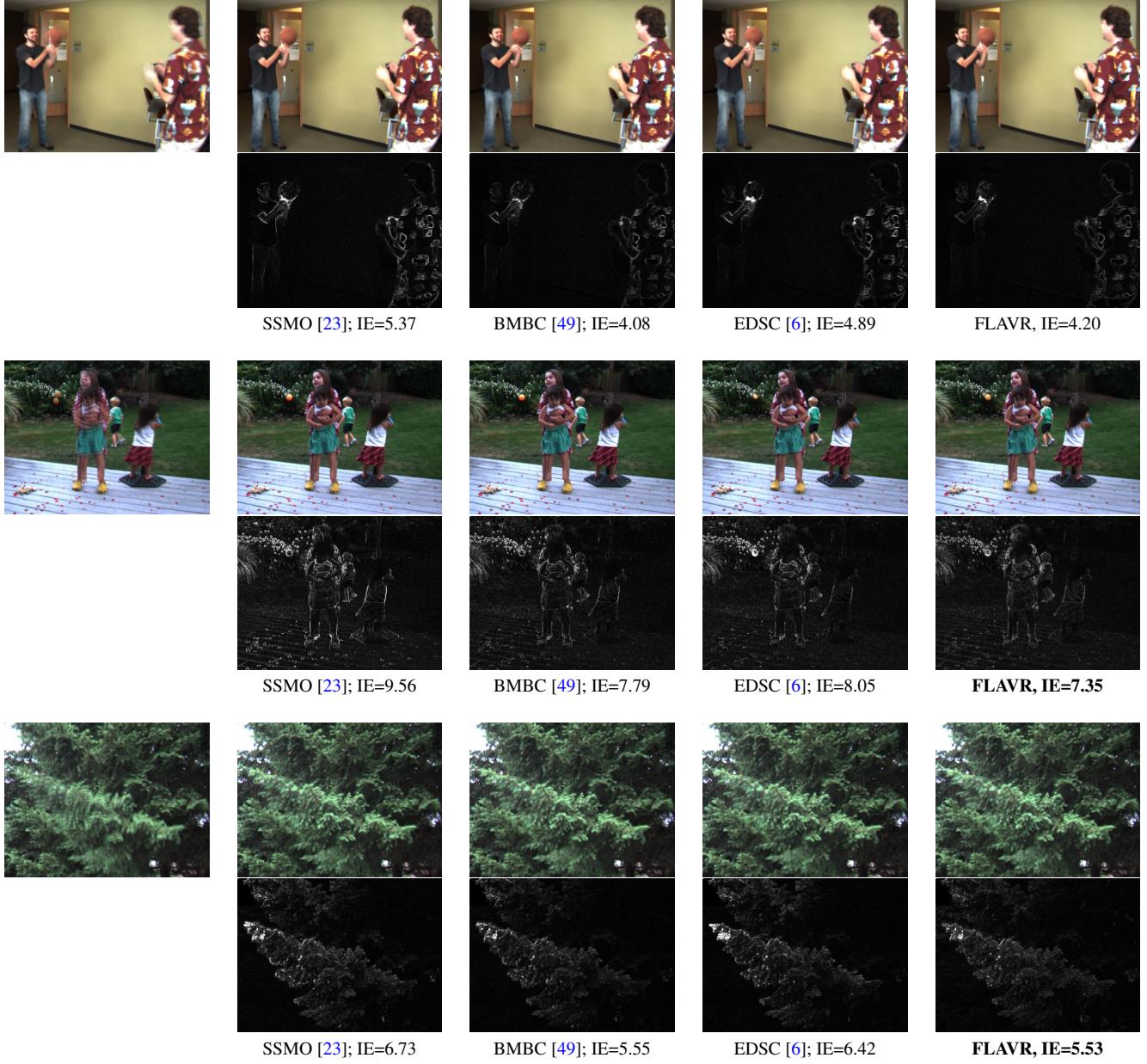


Figure 10. Interpolation results for $2\times$ interpolation on Middlebury test set. The leftmost images in each row represent the overlaid inputs. The first row in each set represents the interpolated frame, while the second row shows the error maps with respect to the ground truth. IE shows the interpolation error of the method. The interpolation errors for all the baselines are reported on the [official leaderboard](#).

- [10] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015. [2](#)
- [11] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 6202–6211, 2019.
- [2](#)
- [12] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3636–3645, 2017. [2](#)
- [13] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision

- benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 8
- [14] Daniel Gordon, Kiana Ehsani, Dieter Fox, and Ali Farhadi. Watching the world go by: Representation learning from unlabeled videos. *arXiv preprint arXiv:2003.07990*, 2020. 2
- [15] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019. 2, 8
- [16] Tengda Han, Weidi Xie, and Andrew Zisserman. Memory-augmented dense predictive coding for video representation learning. *arXiv preprint arXiv:2008.01065*, 2020. 2
- [17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 2
- [18] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 4
- [19] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Rife: Real-time intermediate flow estimation for video frame interpolation. *arXiv preprint arXiv:2011.06294*, 2020. 2, 3
- [20] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017. 8
- [21] Allan Jabri, Andrew Owens, and Alexei A Efros. Space-time correspondence as a contrastive random walk. *arXiv preprint arXiv:2006.14613*, 2020. 8, 10
- [22] Dinesh Jayaraman and Kristen Grauman. Slow and steady feature analysis: higher order temporal coherence in video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3852–3861, 2016. 2
- [23] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9000–9008, 2018. 1, 2, 4, 5, 7, 11, 13
- [24] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 7, 9
- [25] Soomro Khurram, Zamir Amir, and Shah Mubarak. UCF101: A dataset of 101 human action classes from videos in the wild. *CRCV-TR-12-01*, 2012. 4, 11
- [26] Hildegard Kuehne, Hueihan Jhuang, Estibaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *2011 International Conference on Computer Vision*, pages 2556–2563. IEEE, 2011. 11
- [27] Hyeongmin Lee, Taeoh Kim, Tae-young Chung, Daehyun Pak, Yuseok Ban, and Sangyoun Lee. Adacof: Adaptive collaboration of flows for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5316–5325, 2020. 1, 3, 5, 7
- [28] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 667–676, 2017. 2
- [29] Xueting Li, Sifei Liu, Shalini De Mello, Xiaolong Wang, Jan Kautz, and Ming-Hsuan Yang. Joint-task self-supervised learning for temporal correspondence. *arXiv preprint arXiv:1909.11895*, 2019. 8
- [30] Pengpeng Liu, Michael Lyu, Irwin King, and Jia Xu. Selfflow: Self-supervised learning of optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4571–4580, 2019. 8
- [31] Yihao Liu, Liangbin Xie, Li Siyao, Wenxiu Sun, Yu Qiao, and Chao Dong. Enhanced quadratic video interpolation. In *European Conference on Computer Vision*, pages 41–56. Springer, 2020. 2
- [32] Yu-Lun Liu, Yi-Tung Liao, Yen-Yu Lin, and Yung-Yu Chuang. Deep video frame interpolation using cyclic frame generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8794–8802, 2019. 2
- [33] Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4463–4471, 2017. 1, 2, 4, 5, 7, 8
- [34] Zelun Luo, Boya Peng, De-An Huang, Alexandre Alahi, and Li Fei-Fei. Unsupervised learning of long-term motion dynamics for videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2203–2212, 2017. 7, 8
- [35] Dhruv Mahajan, Fu-Chung Huang, Wojciech Matusik, Ravi Ramamoorthi, and Peter Belhumeur. Moving

- gradients: a path-based method for plausible image interpolation. *ACM Transactions on Graphics (TOG)*, 28(3):1–11, 2009. 2
- [36] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 8
- [37] Simone Meyer, Abdelaziz Djelouah, Brian McWilliams, Alexander Sorkine-Hornung, Markus Gross, and Christopher Schroers. Phasenet for video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 498–507, 2018. 1, 2
- [38] Simone Meyer, Oliver Wang, Henning Zimmer, Max Grosse, and Alexander Sorkine-Hornung. Phase-based frame interpolation for video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1410–1418, 2015. 2
- [39] Antoine Miech, Ivan Laptev, and Josef Sivic. Learnable pooling with context gating for video classification. *arXiv preprint arXiv:1706.06905*, 2017. 4
- [40] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, pages 527–544. Springer, 2016. 2
- [41] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3883–3891, 2017. 4
- [42] Simon Niklaus and Feng Liu. Context-aware synthesis for video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1710, 2018. 1, 2
- [43] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5437–5446, 2020. 1, 2, 5
- [44] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive convolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 670–679, 2017. 1, 2
- [45] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 261–270, 2017. 1, 2, 4, 5, 9
- [46] Jim Nilsson and Tomas Akenine-Möller. Understanding ssim. *arXiv preprint arXiv:2006.13846*, 2020. 4
- [47] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016. 4
- [48] Tae-Hyun Oh, Ronnachai Jaroensri, Changil Kim, Mohamed Elgharib, Fr'edo Durand, William T Freeman, and Wojciech Matusik. Learning-based video motion magnification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 633–648, 2018. 8, 10
- [49] Junheum Park, Keunsoo Ko, Chul Lee, and Chang-Su Kim. Bmbe: Bilateral motion estimation with bilateral cost volume for video interpolation. *arXiv preprint arXiv:2007.12622*, 2020. 1, 3, 5, 11, 13
- [50] Deepak Pathak, Ross Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. Learning features by watching objects move. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2701–2710, 2017. 2
- [51] Tomer Peleg, Pablo Szekely, Doron Sabo, and Omry Sendik. Im-net for high resolution video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2398–2407, 2019. 1, 2
- [52] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 724–732, 2016. 4
- [53] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 3
- [54] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *German conference on pattern recognition*, pages 31–42. Springer, 2014. 6, 11
- [55] Zhihao Shi, Xiaohong Liu, Kangdi Shi, Linhui Dai, and Jun Chen. Video interpolation via generalized deformable convolution. *arXiv preprint arXiv:2008.10680*, 2020. 2
- [56] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. CDC: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2
- [57] Li Siyao, Shiyu Zhao, Weijiang Yu, Wenxiu Sun, Dimitris Metaxas, Chen Change Loy, and Ziwei Liu. Deep animation video interpolation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6587–6595, 2021. 2, 3

- [58] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1279–1288, 2017. 4
- [59] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018. 2
- [60] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. 2
- [61] Du Tran, Heng Wang, Matt Feiszli, and Lorenzo Torresani. Video classification with channel-separated convolutional networks. In *ICCV*, 2019. 3, 4, 6
- [62] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. 2, 3
- [63] Stepan Tulyakov, Daniel Gehrig, Stamatis Georgoulis, Julius Erbach, Mathias Gehrig, Yuanyou Li, and Davide Scaramuzza. Time lens: Event-based video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16155–16164, 2021. 2, 3
- [64] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *Advances in neural information processing systems*, pages 613–621, 2016. 7, 8
- [65] Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. Tracking emerges by colorizing videos. In *Proceedings of the European conference on computer vision (ECCV)*, pages 391–408, 2018. 2
- [66] Neal Wadhwa, Michael Rubinstein, Frédéric Durand, and William T. Freeman. Phase-based video motion processing. *ACM Trans. Graph. (Proceedings SIGGRAPH 2013)*, 32(4), 2013. 10
- [67] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2015. 2
- [68] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2566–2576, 2019. 2, 8
- [69] Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman. Learning and using the arrow of time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8052–8060, 2018. 2
- [70] Hao-Yu Wu, Michael Rubinstein, Eugene Shih, John Guttag, Frédéric Durand, and William Freeman. Eulerian video magnification for revealing subtle changes in the world. *ACM transactions on graphics (TOG)*, 31(4):1–8, 2012. 8, 10, 11
- [71] Jonas Wulff and Michael J Black. Temporal interpolation as an unsupervised pretraining task for optical flow estimation. In *German Conference on Pattern Recognition*, pages 567–582. Springer, 2018. 8
- [72] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 305–321, 2018. 2, 4
- [73] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10334–10343, 2019. 2
- [74] Huijuan Xu, Abir Das, and Kate Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2017. 2
- [75] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2
- [76] Jiarui Xu and Xiaolong Wang. Rethinking self-supervised correspondence learning: A video frame-level similarity perspective. *arXiv preprint arXiv:2103.17263*, 2021. 8
- [77] Xiangyu Xu, Li Siyao, Wenxiu Sun, Qian Yin, and Ming-Hsuan Yang. Quadratic video interpolation. In *Advances in Neural Information Processing Systems*, pages 1647–1656, 2019. 1, 2, 3, 4, 5, 7, 11
- [78] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8):1106–1125, 2019. 1, 2, 4
- [79] Zhefei Yu, Houqiang Li, Zhangyang Wang, Zeng Hu, and Chang Wen 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. 2

- [80] Liangzhe Yuan, Yibo Chen, Hantian Liu, Tao Kong, and Jianbo Shi. Zoom-in-to-check: Boosting video interpolation via instance-level discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12183–12191, 2019. [2](#)
- [81] Haoxian Zhang, Yang Zhao, and Ronggang Wang. A flexible recurrent residual pyramid network for video frame interpolation. ICCV, 2019. [2](#)
- [82] Haoxian Zhang, Yang Zhao, and Ronggang Wang. A flexible recurrent residual pyramid network for video frame interpolation. In *European Conference on Computer Vision*, pages 474–491. Springer, 2020. [5](#)