IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 4, 2020, accepted July 16, 2020, date of publication July 21, 2020, date of current version July 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3010846

Residual Learning of Video Frame Interpolation Using Convolutional LSTM

KEITO SUZUKI AND MASAAKI IKEHARA, (Senior Member, IEEE)
Department of Electronics and Electrical Engineering, Keio University, Yokohama 223-8522, Japan

Corresponding author: Keito Suzuki (k_suzuki@tkhm.elec.keio.ac.jp)

ABSTRACT Video frame interpolation aims to generate intermediate frames between the original frames. This produces videos with a higher frame rate and creates smoother motion. Many video frame interpolation methods first estimate the motion vector between the input frames and then synthesizes the intermediate frame based on the motion. However, these methods rely on the accuracy of the motion estimation step and fail to accurately generate the interpolated frame when the estimated motion vectors are inaccurate. Therefore, to avoid the uncertainties caused by motion estimation, this paper proposes a method that implicitly learns the motion between frames and directly generates the intermediate frame. Since two consecutive frames are relatively similar, our method takes the average of these two frames and utilizes residual learning to learn the difference between the average of these frames and the ground truth middle frame. In addition, our method uses Convolutional LSTMs and four input frames to better incorporate spatiotemporal information. We also incorporate attention mechanisms in our model to further enhance the performance. This neural network can be easily trained end to end without difficult to obtain data such as optical flow. Our experimental results show that the proposed method without explicit motion estimation can perform favorably against other state-of-the-art frame interpolation methods. Further ablation studies show the effectiveness of various components in our proposed model.

INDEX TERMS Convolutional LSTM, frame interpolation, video processing.

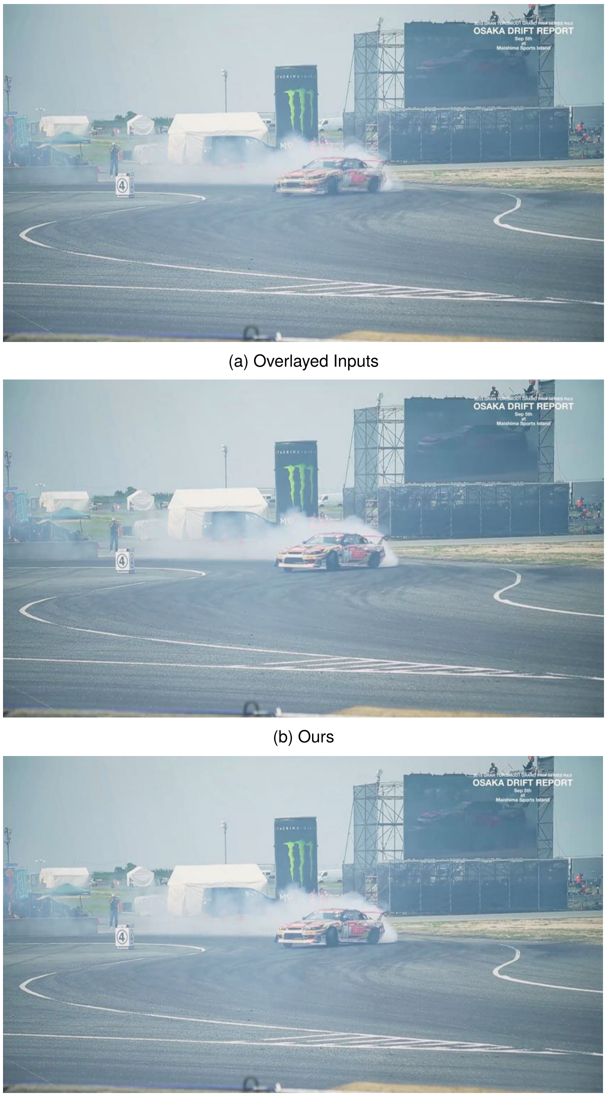
I. INTRODUCTION

The widespread adoption of TVs, computers, and smartphones in everyday life has brought a growing demand for high quality videos. To create clearer videos, super-resolution techniques increase the spatial resolution of low resolution videos. Similarly, video frame interpolation aims to increase the temporal resolution of videos and create more realistic videos by making the motions smoother. Video frame interpolation has gained much attention recently in the field of computer vision and video processing as it has various applications such as frame rate up conversion, frame recovery, and slow motion generation.

Traditional frame interpolation methods are composed of two steps: motion estimation and frame synthesis. In the motion estimation step, typically the pixel wise motion vector or optical flow between two input frames are estimated using gradient based approaches [1]. These approaches rely on the assumption that the brightness of objects remain constant and that the motion is linear between frames. In the frame synthesis step, the intermediate frame is generated based on the estimated optical flow and input frames. However, because frame synthesis is dependent on the optical flow, this approach is not able to reliably create the intermediate frame when the estimated optical flow is inaccurate.

In recent years, deep learning approaches have gained much attention for their success in various computer vision problems such as image recognition [2], style transfer [3], [4], and view synthesis [5]. For video frame interpolation, Long *et al.* [6], first trained a neural network that learns to interpolate frames as an intermediate step for image matching. However, this method tends to create blurry results. Niklaus *et al.* [7], [8]. proposed to combine the flow estimation and frame synthesis steps into a single convolution process. Their method uses a convolutional neural network (CNN) to estimate a spatially adaptive convolution kernel for each pixel location. Although this approach is able to generate pixels from a fixed neighborhood, it requires high memory and computational demand and cannot handle motions that are larger than the specified kernel size.

The associate editor coordinating the review of this manuscript and approving it for publication was Yun Zhang.



(a) Overlayed Inputs

(b) Ours

(c) Ground Truth

FIGURE 1. Example output of video frame interpolation.

The current state-of-the-art methods use CNNs to predict the bidirectional optical flow between input frames and use another model to synthesizes the intermediate frame based on the predicted optical flow. Although the estimated flow is becoming more accurate year by year, it is still difficult to obtain ground truth optical flow, especially for real world sequences. Consequently, flow based methods are difficult to train and the accuracy of the synthesized frame are dependent on the predicted optical flows.

Since both the kernel based and flow based approaches rely on the accuracy of the intermediate step, this paper proposes a simple CNN based approach that directly synthesizes the interpolated frame without explicit motion modeling. However, generating the entire frame from scratch is naive and often results in blurry frames. Since temporally close frames are very similar, it is cumbersome to have a network learn to generate an entire frame. This is especially the case for sequences where there is little to no camera motion and much of the frame remains unchanged. Thus, our model uses residual learning to synthesize a new frame. Specifically, the proposed model takes four consecutive frames as input and learns the difference between the average of the second and third frames and the intermediate frame. This is computationally effective and our experimental results show that residual learning achieves better performance than directly generating the entire frame.

In addition, our model uses Convolutional LSTMs [9] to better incorporate the spatiotemporal information in video sequences. Previous methods use simple convolution layers in their models and does not fully utilize the temporal information that is available in videos. Convolutional LSTM is better able to handle spatial information since it replaces the matrix multiplication in fully connected LSTM with the convolution operation. Convolutional LSTMs have been used in various areas [10]–[12] dealing with videos and can be applied to video frame interpolation. Our experimental results demonstrate that incorporating Convolutional LSTM layers in our model results in a better performance.

This paper is organized as follows. Section II gives a brief overview of the related works in video frame interpolation, Convolutional LSTMs, and attention mechanisms. Section III explains the proposed method in detail. Section IV presents our experimental results compared to state-of-the-art methods on various datasets. Finally, Section V concludes the paper with a brief summary of our contributions.

II. RELATED WORK

A. VIDEO FRAME INTERPOLATION

Video frame interpolation is a classic computer vision problem that has been researched extensively. Conventional methods [13], [14] for frame interpolation typically predicts the pixel-wise motion vectors between two frames using optical flow algorithms and then synthesizes the intermediate frame guided by the estimated motion vectors. The performance of these methods depends on the accuracy of the estimated optical flow which is hindered by issues such as occlusion and large motion.

CNN based approaches have shown to be effective in various computer vision and has achieved state-of-the-art performance in the area of video frame interpolation. Long *et al.* first trained a CNN model to directly generate the intermediate frame for the image matching task. However, their results suffer from blurriness as it is difficult to generate an entire frame from scratch with a rather simple model. Then, Niklaus *et al.* proposed the AdaConv [7] and SepConv [8] which combine the motion estimation and frame synthesis steps from conventional frame interpolation methods into a single convolution process. They use a CNN to estimate a convolution kernel for each pixel location in the input frames and then synthesizes the frame based on the predicted kernels. While this novel approach is easy to train, it requires high memory and heavy computation especially for high resolution sequences.

Much of the research done on frame interpolation in the past few years has been flow based and uses a neural network to estimate the bidirectional optical flow between two frames and then uses another network to create the intermediate frame. Liu *et al*. [15] proposed the deep voxel flow model which synthesizes frame by flowing existing pixels from existing frames. Though this method can be trained without ground truth optical flow data, it still faces difficulties when faced with large motion or when the flows are inaccurate. Liu *et al*. based their CyclicGen [16] model on the voxel flow model but introduces a cycle consistency loss which has been used in tasks such as machine translation and style transfer [17]. They propose that they can produce higher quality frames if the generated frames can be used to accurately reconstruct the input frames. Despite the introduction of a novel loss function and increase in performance, it still faces similar issues with the original voxel flow model.

Bao *et al*. integrates both the kernel based and flow based approach in their MEMC-Net [18]. They introduce an adaptive warping layer that takes the estimated optical flows and interpolation kernels as input to warp the input frames. Since these existing methods implicitly handle occlusion, Bao *et al*. proposes the DAIN [18] model which incorporates a depth estimation network to explicitly handle the occlusion. They introduce a depth-aware flow projection layer which puts more weight on closer objects when sampling from the frames. DAIN uses pretrained models for the optical flow estimation network and the depth estimation network, and only requires video sequences to train. However, the model still does not perform well when the predicted flows are inaccurate.

Our method does not rely on an intermediate step such as kernel estimation or optical flow estimation as there are uncertainties involved in these steps. We propose a CNN based direct synthesis method similar to Long *et al.*, but does not generate an entire frame from scratch. It learns the residual between the average of two input frames and the intermediate frame and can effectively synthesize clear frames. The proposed approach can be trained end to end without hard to obtain data such as optical flow.

B. RESIDUAL LEARNING

Residual learning was first proposed by He *et al.* [19] in their ResNet model for the image recognition task. They overcame the vanishing gradient problem in deep neural networks by using residual units with identity mappings as skip connections. Residual learning has also been used in many image restoration tasks for the network to learn the residual or difference between the corrupted image and clean image. For example, in the image denoising task, a CNN is often used to learn the noise component which is then subtracted from the noisy image to output a clean image. This helps ease the training since the CNN does not have to learn to generate an entire image from scratch.

In our case, residual learning is used in both the frame synthesis network and the refinement network. In the frame synthesis network, the network learns the difference between the average of two consecutive frames and the intermediate frame. In the refinement network, the network learns the difference between the frame created by the frame synthesis network and the intermediate frame to further refine the image.

C. CONVOLUTIONAL LSTM

Long Short Term Memory (LSTM) [20] is an architecture of RNN that is capable of learning both long and short term dependencies. In traditional LSTM structures, the hidden states are calculated using vector multiplication. Therefore, an LSTM layer can only take a fixed sized vector input and disregards any spatial information in data such as images. To apply the LSTM structure for spatiotemporal data, the Convolutional LSTM (ConvLSTM) [9] structure was proposed. It was first used for precipitation nowcasting where it took previously recorded radar echo sequences as input to forecast future radar maps. Since then, ConvLSTM has been used in a variety of tasks such as depth estimation [11], [12] and cell segmentation [10].

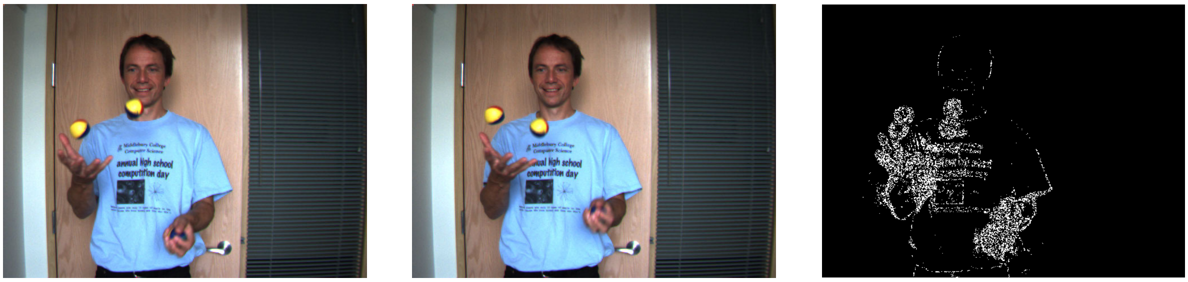
Our method incorporates ConvLSTMs in the encoding layers of the U-Net model following [10]–[12]. The model takes four consecutive frames as input and synthesizes the frame between the second and third input frames. By using ConvLSTMs, the network is better able to encode the spatiotemporal information that are present in video sequences.

D. ATTENTION MECHANISM

Recently attention mechanisms have been a popular and powerful method to enhance deep neural networks. It is based on human perception that we pay visual attention to a certain region when processing a large amount of information. Attention mechanism was first proposed in the field of natural language processing [21] to attend to certain parts of the source input in order to learn the relationship between the source and target.

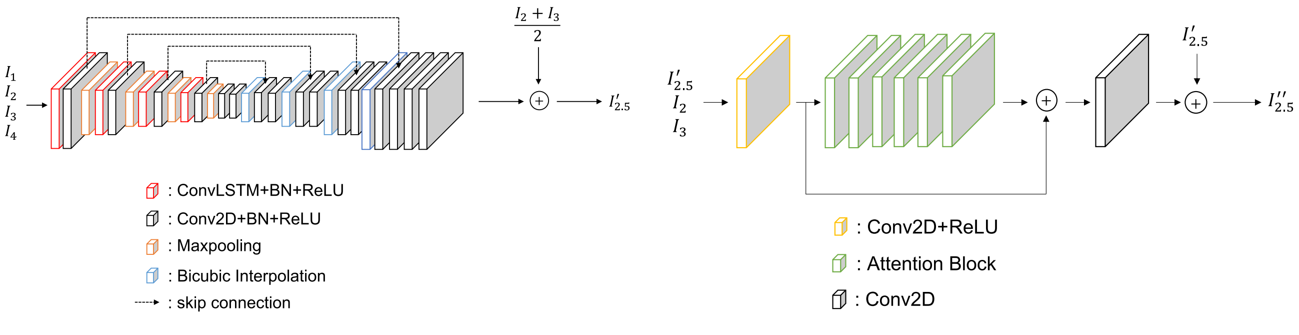
Since then, attention mechanisms with specialized architectures have been proposed for various computer vision tasks. Wang *et al.* [22] proposed the residual attention network which stacks attention modules for image classification. Hu *et al.* [23] proposed the squeeze-and-excitation (SE) block that learns the channel interdependencies to recalibrate the channel-wise features to improve the image classification accuracy. The SE block has been incorporated in the super-resolution task [24], [25] where [25] extends the module by adding a spatial attention mechanism.

Our method uses attention modules to refine the interpolated frame. The module that we use is nearly identical to [25] and has both channel and spatial attention but for the channel attention our module only conducts global average pooling on the input features. By incorporating attention mechanisms, our model is able to better reconstruct the fine details in intermediate frames.



(a) Frame 1 (b) Frame 2 (c) Difference

FIGURE 2. Difference between two consecutive frames.



(a) Frame Synthesis Network (b) Refinement Network

FIGURE 3. Overall architecture of the proposed model.

III. PROPOSED METHOD

A. NETWORK STRUCTURE

The architecture of the network is shown in Figure 3. The model is split into two parts: the frame synthesis network and the refinement network.

1) FRAME SYNTHESIS NETWORK

In this network, four consecutive frames are used as input and the goal is to synthesize the frame that comes in between the second and the third frame. In previous methods, an explicit motion estimation step was necessary to interpolate a frame, but this network is able to implicitly learn motion between frames. The easiest frame interpolation method is the linear interpolation where the average of the two frames is computed, but this results in blurry artifacts. Hence, to create a much clearer frame, the proposed network learns the residual of the ground truth frame and the average between the second and third frame. Temporally close frames are relatively similar as seen in Fig. 2. so therefore, residual learning is effective as the model only has to learn to readjust regions where there is motion rather than generate the entire frame. The output of this part of the model can be expressed as:

$$I'_{2.5} = \frac{I_2 + I_3}{2} + f_1(I_1, I_2, I_3, I_4) \tag{1}$$

where I_1, I_2, I_3, I_4 are the four input frames, $I'_{2.5}$ is the interpolated frame between I_2 and I_3, and $f_1(x)$ is the function modeled by the frame synthesis network.

This network takes the form of a U-Net [26] structure. In the encoding portion, a Maxpooling layer followed by a ConvLSTM layer and a Conv2D layer is used. Interweaving recurrent units into the encoder allows the network to learn not only the spatial but also the temporal information. Bidirectional ConvLSTM units are used to be able to encode both past and future frames. In the decoding phase, a bicubic interpolation followed by two Conv2D layer is used to upsample the tensor to its original image size. The placement of ConvLSTM units only in the encoding phase is inspired by the works of [10]–[12] where they showed that placing ConvLSTMs just in the encoding phase led to better results than placing them just in the decoding phase or throughout the entire network.

2) REFINEMENT NETWORK

In this network, the synthesized frame $I'_{2.5}$ as well as its adjacent frames I_2 and I_3 are fed as input to produce the final intermediate frame $I''_{2.5}$. This refinement network has the goal of refining the details in the frame. We incorporate the attention mechanism into this network following its success in various fields. The network uses multiple Attention Blocks which are inspired from the works in super-resolution [24], [25]. The architecture of the Attention Block is shown in Figure 4 and uses a sequence of channel and spatial attention.

Channel attention is used to exploit the channel dependencies within the features. We use the same model as the SE block [23] and the detailed structure can be seen in Figure 5a. Given an input feature F_{in}, we first apply global average

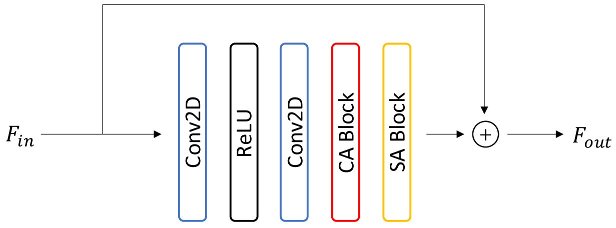


FIGURE 4. Architecture of the attention block.

pooling to aggregate the spatial information of a feature map. This is then followed by two 1 × 1 Conv2D layers each followed by ReLU and Sigmoid activation functions respectively to obtain the channel attention map. The final output feature F_{out} is obtained by element-wise multiplication of the input feature and the estimated channel attention map. This operation can be expressed with the following equation:

$$A_c = \sigma(W_1 * \delta(W_0 * AvgPool(F_{in}))) \tag{2}$$

where W_0 and W_1 are the two 1 × 1 convolution layers, and δ and σ denote the ReLU and Sigmoid activation functions.

Spatial attention is used to learn where the important locations are in features. We use the spatial attention module that was used in [25] and the detailed structure can be seen in Figure 5b. Given an input feature F_{in}, we apply both average pooling and max pooling over the channel axis and concatenate them to aggregate the channel information. Then, the spatial attention map is generated by applying a single Conv2D layer followed by a Sigmoid activation function. The final output feature F_{out} is obtained by element-wise multiplication of the input feature and the generated spatial attention map. The spatial attention can be expressed with the following equation:

$$A_s = \sigma(W_2 * [MaxPool(F_{in}), AvgPool(F_{in})]) \tag{3}$$

where W_2 is a 7 × 7 convolution layer and σ is the Sigmoid activation function.

While this network may seem redundant and unnecessary, experimental results show that this network helps to generate clearer images. The proposed network adopts a residual structure to refine the image produced from the first network. The output image can be expressed as:

$$I''_{2.5} = I'_{2.5} + f_2(I'_{2.5}, I_1, I_2) \tag{4}$$

where $I''_{2.5}$ is the final intermediate frame and $f_2(x)$ is the function modeled by the refinement network.

B. LOSS FUNCTION

The model is trained by optimizing the following loss function:

$$\mathcal{L} = \lambda \mathcal{L}_1 + \mathcal{L}_2 \tag{5}$$

$$\mathcal{L}_1 = \sum_{i=1} \|I'_{i,2.5} - I_{i,2.5}\|_1 \tag{6}$$

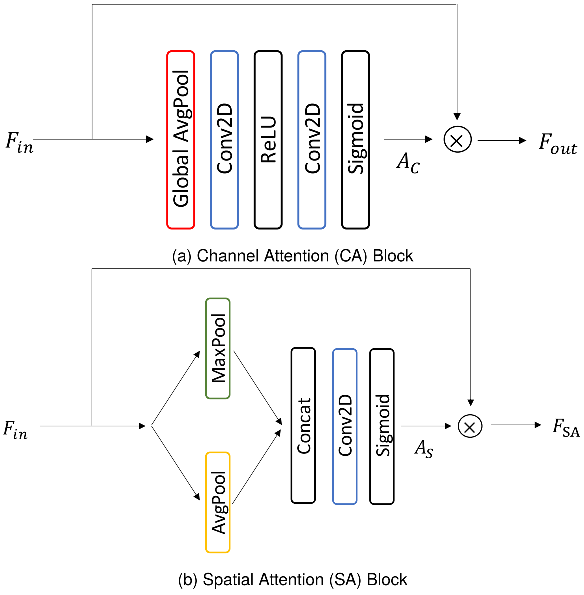


FIGURE 5. The architecture of the attention modules.

$$\mathcal{L}_2 = \sum_{i=1} \|I''_{i,2.5} - I_{i,2.5}\|_1 \tag{7}$$

where $I_{2.5}$ is the ground truth frame. The L_1 norm of both $I'_{2.5}$ and $I''_{2.5}$ is taken with the ground truth frame. The weight λ was determined empirically and set to 0.5. This forces the frame synthesis network to generate a frame that resembles the ground truth frame rather than create an arbitrary feature map.

C. TRAINING

The model is trained using the Vimeo90K Septuplet dataset [27] which contains seven consecutive frames extracted from 39K selected video clips. There are 64,612 7-frame sequences for the training dataset with a fixed resolution of 448 × 256. The first, third, fifth, and seventh frames are used as input to predict the fourth frame in the sequence.

For data augmentation, the frames are first randomly cropped to a size of 256 × 256. In addition, the training data is augmented by horizontal flipping as well as reversing the temporal order of the sequence. The pixel values of the frames are also normalized to a range of [-1, 1]. The optimization of the proposed network is done with the Adam [28] optimizer where B1 and B2 are set to 0.9 and 0.999. The learning rate is set to 0.0001, while the batch size is set to 4. The entire model is trained for 150 epochs with the PyTorch framework on a single NVIDIA GeForce GTX 1080 Ti graphics card which takes about seven days to converge.

IV. EXPERIMENTAL RESULTS

In this section, we test the proposed network on several datasets including Vimeo90K [27], Middlebury [29], and DAVIS 2017 [30]. The proposed method is quantitatively and qualitatively compared to the kernel based frame

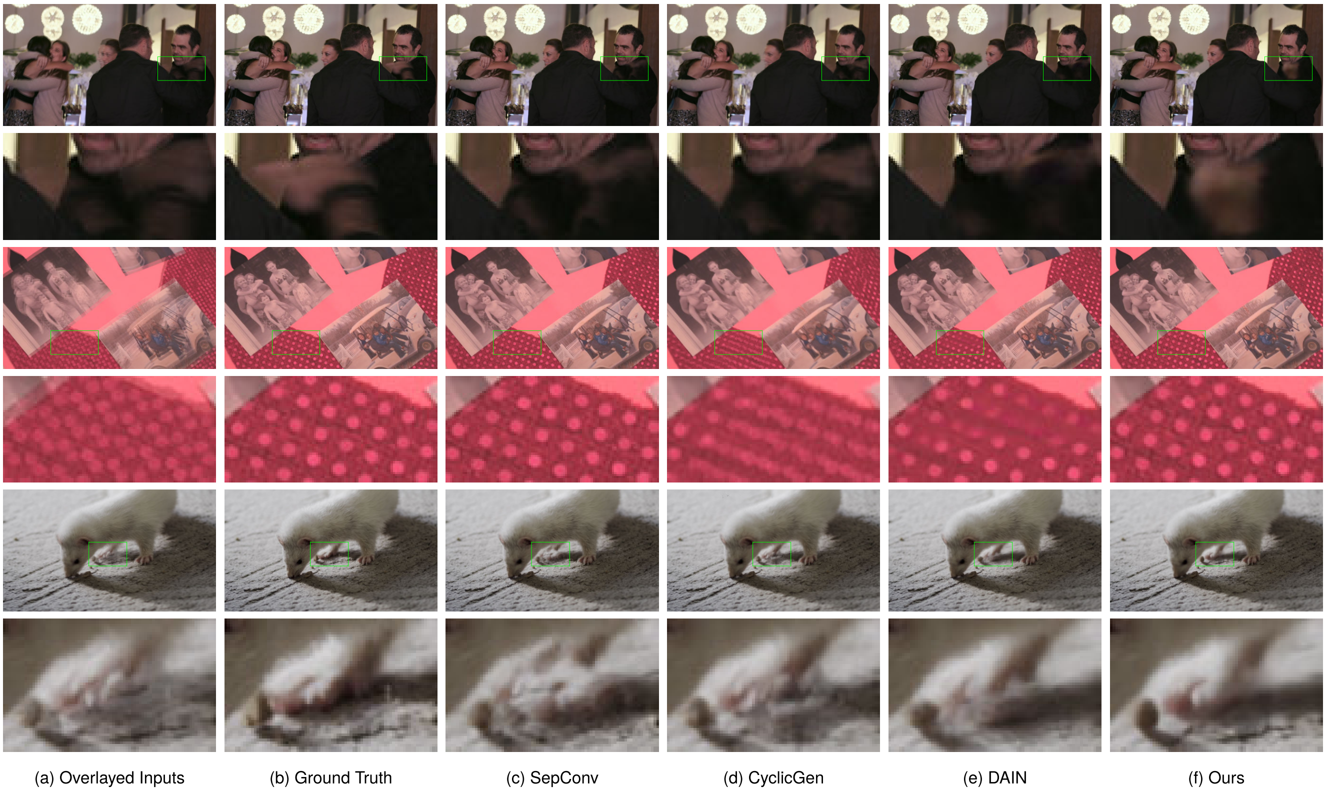


FIGURE 6. Visual comparison on the Vimeo90K dataset.

interpolation method SepConv [8] and flow based methods CyclicGen [16] and DAIN [18]. In addition, ablation studies are

conducted to assess the effectiveness of the various components of our network.

A. DATASETS

The following datasets are used to compare the proposed method to state-of-the-art frame interpolation methods. These methods are compared quantitatively using PSNR and SSIM values and then compared qualitatively.

1) Vimeo90K

There are 7,824 septuplets in the test set of the Vimeo90K dataset. The sequences are extracted from real world video sequences and is similar to the training dataset for the proposed model. The image resolution in this dataset is 448 × 256.

2) MIDDLEBURY

The Middlebury benchmark is widely used to evaluate optical flow and video frame interpolation. There are two subsets of the dataset: Evaluation and Other. While the Evaluation dataset is used to officially rank various frame interpolation methods, the ground truth intermediate frame is not publicly available. As a result, the Middlebury Other dataset is used in this experiment to qualitatively and quantitatively compare the results. The image resolution in this dataset is around 640 × 480 pixels.

TABLE 1. Quantitative comparison on the Vimeo90K dataset.

Method	Vimeo90K		DAVIS 2017	
	PSNR	SSIM	PSNR	SSIM
SepConv [8]	33.38	0.939	27.73	0.864
CyclicGen [16]	32.08	0.916	28.44	0.880
DAIN [18]	34.57	0.952	28.23	0.873
Ours	35.65	0.956	28.86	0.886

3) DAVIS2017

The DAVIS 2017 dataset [30] is a publicly available dataset that is used to benchmark video object segmentation. It consists of many video sequences on various real life scenes. The sequences from this dataset with little to no camera motion are used to evaluate the proposed method. The odd frames are used as input to predict the even frames. The size of the frames vary among the video sequences, but all have 480p resolution.

B. RESULTS

We show the evaluation of our method on the Vimeo90K dataset [27] in Table 1. In terms of both PSNR and SSIM, our proposed method achieves the best average performance. All of the sequences in this dataset are from real scenes and therefore shows that our model performs well in these

TABLE 2. Quantitative comparison on the Middlebury Other dataset.

Videos	SepConv [8]		CyclicGen [16]		DAIN [31]		Ours	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
DogDance	30.34	0.894	31.46	0.917	32.46	0.927	34.13	0.942
Minicooper	30.41	0.967	30.68	0.963	31.6	0.971	32.24	0.973
RubberWhale	45.25	0.992	42.43	0.985	44.09	0.990	44.19	0.988
Walking	31.02	0.940	33.58	0.953	34.25	0.960	36.86	0.966
Beanbags	29.76	0.926	28.19	0.913	32.84	0.937	31.29	0.933
Hydrangea	36.66	0.984	36.02	0.976	37.42	0.985	37.58	0.982
Sub-Average	33.91	0.951	33.73	0.951	35.44	0.961	36.05	0.964
Grove2	36.45	0.979	32.20	0.921	36.85	0.980	36.37	0.978
Grove3	30.95	0.955	27.93	0.876	31.20	0.955	30.71	0.947
Urban2	41.1	0.984	29.58	0.829	43.62	0.989	33.13	0.947
Urban3	40.56	0.981	32.58	0.880	39.87	0.984	38.15	0.955
Total Average	35.25	0.960	32.45	0.921	36.42	0.968	35.46	0.958

sequences. A visual comparison of the interpolated frames from this dataset is shown in Figure 6. In the top image of Figure 6, it is evident that our method is best able to reconstruct the moving hand. Other methods result in a ghosting artifact where the output image is similar to the overlayed inputs and cannot accurately locate where the hand should be. In contrast, the interpolated frame from our method is better able to locate where the hand should go and is more consistent with the ground truth image. In the middle image of Figure 6, the two flow based methods, CyclicGen [16] and DAIN [31], are not able to clearly reconstruct the repeating patterns of the rug. This is most likely due to the fact that it is difficult to estimate the flow when there are many similar patterns nearby an object. On the contrary, our direct method and the kernel based SepConv [8] can accurately synthesize these patterns. In the bottom image, our method interpolates the paw more clearly than the other methods.

In Table 2, we evaluate our method on the Middlebury Other benchmark [29]. Although there are originally 12 sequences, we assess our model on 10 sequences that have four input frames. Our method achieves the second best average PSNR score in comparison to the other three methods. In this table, the first 6 sequences are from real scenes while the bottom 4 sequences are synthetic sequences from animations. On the sequences of real scenes, our method achieves the highest average PSNR and SSIM values. However, on the synthetic sequences our method is not able to score as well. Since our model was trained on a dataset with only real scenes, it is inevitable that it does not perform as well on the synthetic video sequences. Our method performs especially poor on the Urban2 sequence with a 10.49 dB difference in terms of PSNR when compared to DAIN [31]. A visual comparison of this sequence is shown in Figure 7. As can be seen, DAIN [31] is able to recover the sharp edges on the side of the building while our method results in blurred edges.

In order to test our method on sequences with little camera motion, we use selected video sequences from the DAVIS 2017 dataset [30]. The results are shown in Table 1. These sequences have little to no camera motion and shows that our method is able to perform the best among the other methods in this sort of situation. This is the effect of residual learning. When there is no camera motion, the network only has to change the areas with motion while keeping the majority of the frame the same. In contrast, when there is camera motion, the entire frame shifts and most of the pixel values change between frames. Consequently, it is easier for our network to synthesize frames in which there is no camera motion.

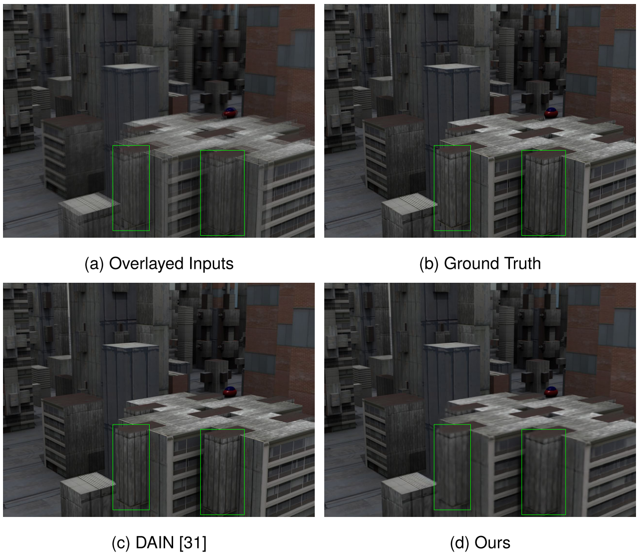

(a) Overlayed Inputs
(b) Ground Truth
(c) DAIN [31]
(d) Ours

FIGURE 7. Visual comparison of Urban2 sequence from Middlebury Other dataset.

C. ABLATION STUDY

We conducted the following ablation studies of our proposed model. Each model in this section is trained with 1/10 of the Vimeo90K Septuplet training set with the same training parameters as our full proposed model.

1) EFFECT OF OUR MODEL COMPONENTS

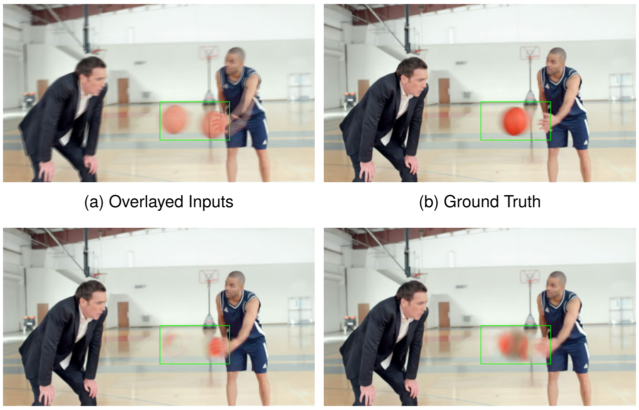
We first evaluated the effectiveness of the major components in our proposed model. The results are shown in Table 3. We started our analysis with a baseline model without ConvLSTMs that generates a frame from scratch. Then, we incorporated ConvLSTMs to the network, followed by the

TABLE 3. Evaluation of our model structure. We start with a baseline UNet model and then sequentially add ConvLSTM, residual structure, and refinement network to get our proposed model.

Model Structure	Baseline	+ConvLSTM	+ConvLSTM +Residual	Full Model
PSNR	30.66	32.04	33.84	34.57
SSIM	0.883	0.919	0.940	0.947

TABLE 4. Effectiveness of using four input frames and incorporating attention mechanism.

Model Structure	2 frames w/ Attention	4 frames w/o Attention	4 frames w/ Attention
PSNR	33.69	34.53	34.57
SSIM	0.940	0.946	0.947


(a) Overlayed Inputs

(b) Ground Truth

(c) Using 2 Frames

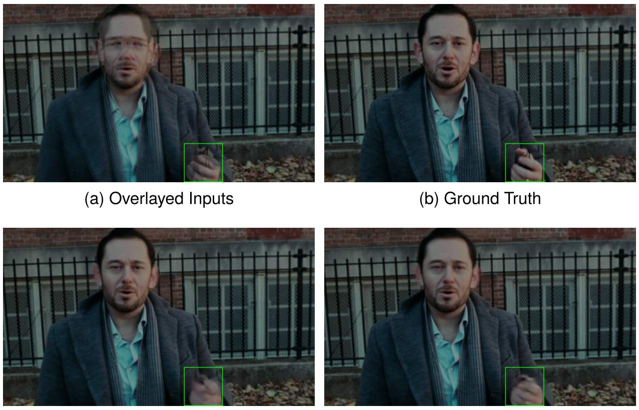
(d) Using 4 Frames

FIGURE 8. Visual comparison of using 2 frames vs. 4 frames as input.

residual of the input frames, and finally added a refinement network with attention mechanisms to complete our model. The results show that our full model with all three components performs the best.

2) EFFECT OF USING 4 FRAMES

Most frame interpolation methods use 2 consecutive frames as input to generate an intermediate frame. These methods rely on the underlying assumption that the motion between 2 frames are linear and hence objects that move in betweeen frames are interpolated to be in a position that is in the middle of the 2 input locations. While this assumption may be true, it is not always accurate especially for low frame rate videos where there is large motion between frames. Our proposed method uses 4 input frames to more accurately determine the interpolated frame. The quantitative comparison is shown in Table 4 where the model that takes 4 input frames is better in terms of both PSNR and SSIM. A qualitative comparison is shown in Figure 8. When using only 2 frames as input, the learned model is unsure about where to place the basketball in the interpolated frame and thus results in imprecise interpolation. On the other hand, when using 4 input frames the model can more accurately predict where the ball should be in the intermediate frame.


(a) Overlayed Inputs

(b) Ground Truth

(c) Without Attention

(d) With Attention

FIGURE 9. Visual comparison of using attention mechanisms.

3) EFFECT OF THE ATTENTION MECHANISM

We tested the effect of incorporating attention mechanism in the refinement network. The numerical results are shown in Table 4 where the attention blocks are able to increase the performance in terms of both PSNR and SSIM. A visual comparison is shown in Figure 9. Since the model is trained on only 1/10 of the full training data, the results are still blurry but using attention blocks helps to recover the fingers in the moving hand. It is presumed that the attention mechanisms help to suppress the low frequency information in the interpolated frame and focus more on reconstructing the high frequency information.These results indicate the effectiveness of integrating attention blocks in our refinement network.

V. CONCLUSION

In this paper, we proposed a simple yet effective video frame interpolation method using residual learning. The proposed architecture is able to incorporate the spatiotemporal information from four consecutive frames and directly generates a realistic intermediate frame without an explicit motion estimation step. In addition, the use of residual learning greatly improved the accuracy of the interpolation since the network does not have to create an image from scratch. We also incorporated attention mechanisms in our model to further improve the performance. The presented method is able to compete against state of the art methods in real world video sequences, especially those with little camera motion.

REFERENCES

[1] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, nos. 1–3, pp. 185–203, Aug. 1981.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[3] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2414–2423.

[4] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 694–711.

[5] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros, "View synthesis by appearance flow," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 286–301.

[6] G. Long, L. Kneip, J. M. Alvarez, H. Li, X. Zhang, and Q. Yu, "Learning image matching by simply watching video," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 434–450.

[7] S. Niklaus, L. Mai, and F. Liu, "Video frame interpolation via adaptive convolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 670–679.

[8] S. Niklaus, L. Mai, and F. Liu, "Video frame interpolation via adaptive separable convolution," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 261–270.

[9] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 802–810.

[10] A. Arbelle and T. R. Raviv, "Microscopy cell segmentation via convolutional LSTM networks," in *Proc. IEEE 16th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2019, pp. 1008–1012.

[11] R. Wang, S. M. Pizer, and J.-M. Frahm, "Recurrent neural network for (Un-) supervised learning of monocular video visual odometry and depth," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 5555–5564.

[12] A. C. S. Kumar, S. M. Bhandarkar, and M. Prasad, "DepthNet: A recurrent neural network architecture for monocular depth prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 283–291.

[13] J. He, G. Yang, X. Liu, and X. Ding, "Spatio-temporal Saliency-based Motion Vector Refinement for Frame Rate Up-conversion," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 16, no. 2, pp. 1–18, 2020.

[14] J. He, G. Yang, J. Song, X. Ding, and R. Li, "Hierarchical prediction-based motion vector refinement for video frame-rate up-conversion," *J. Real-Time Image Process.*, vol. 17, pp. 259–273, Mar. 2018.

[15] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, and A. Agarwala, "Video frame synthesis using deep voxel flow," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4463–4471.

[16] Y.-L. Liu, Y.-T. Liao, Y.-Y. Lin, and Y.-Y. Chuang, "Deep video frame interpolation using cyclic frame generation," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 1–9.

[17] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2223–2232.

[18] W. Bao, W.-S. Lai, X. Zhang, Z. Gao, and M.-H. Yang, "MEMC-net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Sep. 17, 2019, doi: 10.1109/TPAMI.2019.2941941.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[21] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[22] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3156–3164.

[23] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[24] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 286–301.

[25] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.

[26] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241.

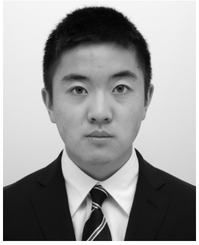
[27] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *Int. J. Comput. Vis.*, vol. 127, no. 8, pp. 1106–1125, Aug. 2019.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

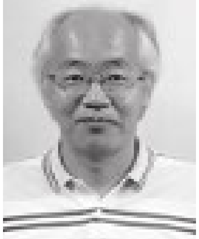
[29] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *Int. J. Comput. Vis.*, vol. 92, no. 1, pp. 1–31, Mar. 2011.

[30] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool, "The 2017 DAVIS challenge on video object segmentation," 2017, *arXiv:1704.00675*. [Online]. Available: http://arxiv.org/abs/1704.00675

[31] W. Bao, W.-S. Lai, C. Ma, X. Zhang, Z. Gao, and M.-H. Yang, "Depth-aware video frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3703–3712.



KEITO SUZUKI received the B.E. degree in electronics and electrical engineering from Keio University, Yokohama, Japan, in 2020. He is currently pursuing the M.E. degree with Keio University under the supervision of Prof. Masaaki Ikehara. His research interests include applying deep learning to video frame interpolation and video super resolution.



MASAAKI IKEHARA (Senior Member, IEEE) received the B.E., M.E., and Dr.Eng. degrees in electrical engineering from Keio University, Yokohama, Japan, in 1984, 1986, and 1989, respectively. From 1989 to 1992, he was an Appointed Lecturer with Nagasaki University, Nagasaki, Japan. He joined the Faculty of Engineering, Keio University, in 1992. From 1996 to 1998, he was a Visiting Researcher with the University of Wisconsin, Madison, and Boston University, Boston, MA, USA. He is currently a Full Professor with the Department of Electronics and Electrical Engineering, Keio University. His research interests include multi-rate signal processing, wavelet image coding, and filter design problems.