RVFIT: Real-time Video Frame Interpolation Transformer

Linlin Ou[1] Yuanping Chen[2] *

[1] [2]Computer Network Information Center of Chinese Acadamy of Science, Beijing, China

[1] University of Chinese Acadamy of Science

ABSTRACT

Video frame interpolation (VFI), which aims to synthesize predictive frames from bidirectional historical references, has made remarkable progress with the development of deep convolutional neural networks (CNNs) over the past years. Existing CNNs generally face challenges in handing large motions due to the locality of convolution operations, resulting in a slow inference structure. We introduce a Real-time video frame interpolation transformer (RVFIT), a novel framework to overcome this limitation. Unlike traditional methods based on CNNs, this paper does not process video frames separately with different network modules in the spatial domain but batches adjacent frames through a single UNet-style structure end-to-end Transformer network architecture. Moreover, this paper creatively sets up two-stage interpolation sampling before and after the end-to-end network to maximize the performance of the traditional CV algorithm. The experimental results show that compared with SOTA TMNet [1], RVSRT has only 50% of the network size (6.2M vs 12.3M, parameters) while ensuring comparable performance, and the speed is increased by 80% (26.1 fps vs 14.3 fps, frame size is 720*576).

Keywords: Video frame interpolation, vision transformer, deep learning.

1. INTRODUCTION

Video frame interpolation (VFI) is a fundamental video enhancement task in which intermediate frames are generated between consecutive ones to increase the frame rate. It is effective in mitigating motion judder and blur and has become an essential strategy for multiple applications, such as video compression [2], video restoration [3] [4] [5], view synthesis [6] [7] and slow-motion generation [8] [9] [10] [11] [12] [13]. Other universal algorithms applied optical flow warping to process this challenging task.

Recently, computer vision [14] [15] [16] and natural language processing [17] [18] [19](NLP) tasks have achieved remarkable progress by using Transformers, which is a remarkably adaptive architecture with advanced modelling capability. Since the vanilla Transformer needs high memory and computational cost, we introduced UNet [20] architecture to utilise features processed in different scales, enlarging the receptive field and reducing the computational complexity. Moreover, to reduce the quadratic complexity [21] [22] [16] [23], the core module of RVFIT is built upon window-based attention by which feature maps are divided into non-overlapping window chips, and self-attention is only performed within each sub-window. Such an approach prohibits information interaction between different windows and leads to a limited receptive field.

In this paper, we are inspired to explore the application of Transformers in video frame interpolation and introduce a superior neural network RVFIT. This approach leads to a much lighter-weight network than the previous methods and a real-time inference speed without sacrificing much performance, as shown in Figure 1. Specifically, our contributions as summarized as follows:

- Inside RVFIT, we design a cascaded Encoder-Decoder architecture to incorporate partial temporal and spatial information for synthesizing high-resolution videos. In particular, we have cleverly combined the first and second versions of Swin-Transformer to make it perform at its best for the first time.
- We propose three RVFIT models with architectures with different complexity, resulting in small (S), medium (M), and large (L) architectures. Experiments show that RVFIT is remarkably faster and smaller than the SOTA VFI methods while maintaining similar performance:

* Further author information:
Linlin Ou: Email: oulinlin@cnic.cn
Yuanping Chen: Email: ypchen@cashq.ac.cn

- RVFIT-L performs similarly to TMNet [1] with 18% fewer parameters, RVFIT-M outperforms Zooming SlowMo [5] with 8% fewer parameters, and RVFIT-S outperforms STARNet [3] with 45% fewer parameters.

- RVFIT-S achieves a frame rate of more than 24 per second (the standard cinematic frame rate) on 720 * 576 frames. It achieves the Zooming SlowMo [5] with a 75% speedup and outperforms STARNet [3] with around 700% speedup.

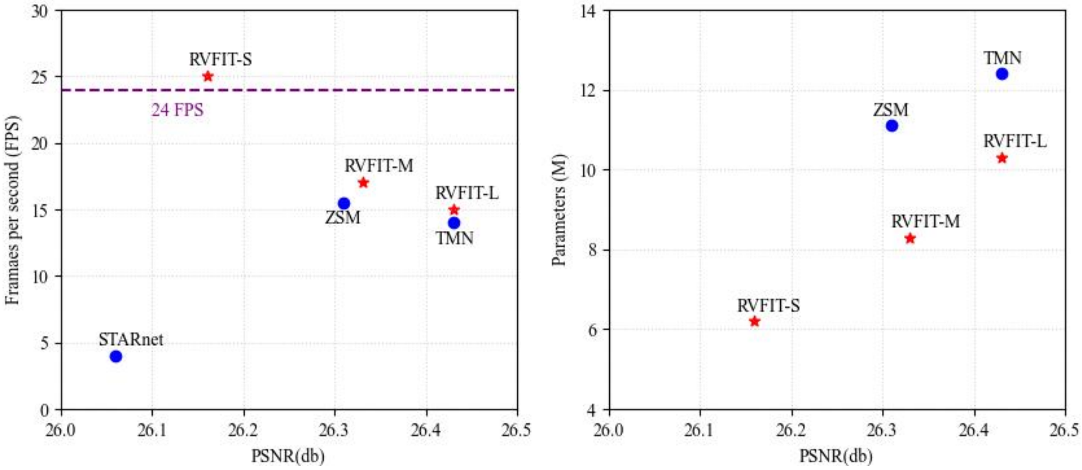

Figure 1. Performance of RVFIT on Vid4 dataset [24]] using small (S), medium (M) and large (L) architectures compared to other baseline models. The upper sub-figure displays FPS versus PSNR. Espacially 24 FPS is the standard cinematic frame rate [25] . The lower sub-figure presents parameter number (in millions) versus PSNR.

2. RELATED WORKS

2.1 Video frame interpolation

Existing video frame interpolation methods can be generally divide three categories: kernel-based [26] [12] [27], flow-based [8] [9] [28] [29] [30], and direct-regression-based methods [31] .

The kernel-based methods [26] [12] [27] do not rely on any prescribed assumptions and thus synthesis better to diverse videos. For example, AdaCoF [26] learns deformable spatially-variant kernels proposed to convolve with the input frame sequence to generate the target frame, and SepConv [12] predicts adaptive separable kernels to aggregate source pixels of the input.

Unlike the kernel-based methods, the flow-based methods [8] [9] [28] [29] [30] produce intermediate frames by warping pixels from the source images according to predicted optical flow. Though these approaches perform well, they usually rely on simplified motion assumptions such as linear [9] and quadratic [32], so they underfit real scenarios.

However, these methods usually apply the kernel prediction modules at one scale and cannot effectively handle complex motions on various scales. Besides, these CNN-based methods cannot account for long-term dependency among pixels. Conversely, we designed a multi-scale Transformer-based kernel generative model, which achieves high-quality results.

2.2 Vision Transformer

Transformer [19] is an overall architecture in Natural Language Processing (NLP) and achieves SOTA performance in various tasks [18] [17]. Lately, Transformers have become popular in the CV field. The advanced Vision Transformer (ViT) [15] computes attention between flattened image patches to solve image classification tasks and exceeds CNN-based methods. TTSR [33] introduced a texture Transformer in low-level vision to find relevant texture patches from referring image to the input image. Because of the high computational costs of attention calculation on videos, these methods only aggregate information on the narrow temporal window. Liu et al. [16] [34] propose a novel transformer-based backbone for vision tasks, Shifted window (Swin) Transformer, to decrease computational complexity by limiting the attention computations inside local and later shifted local windows. Afterwards, [23] designed a U-shape network based on Swin Transformer for general image restoration. SwinIR [22] used Swin Transformer to handle the image restoration task and proposed residual Swin Transformer blocks. RSTT [35] built a spatial-temporal transformer that naturally incorporates the spatial and temporal super-resolution modules into a single model.

3. THE PROPOSED METHOD

In this section, we first give an overview of RVFIT in Section 3.1. Then we discuss the encoder and decoder part of our proposed method in Section 3.2 and Section 3.3, respectively. Finally, the training details are shown in Section 3.4.

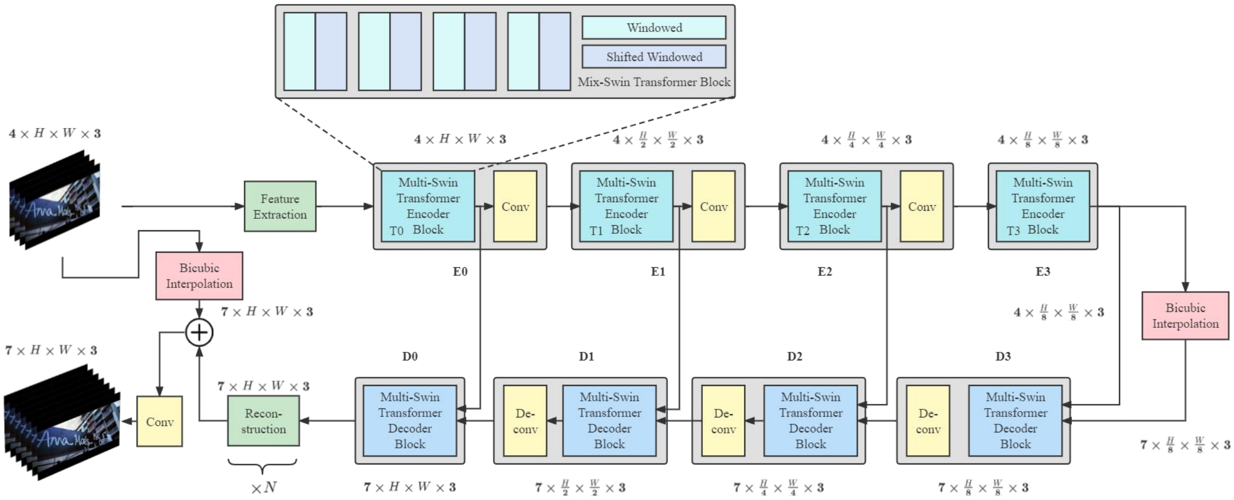
Windowed
Shifted Windowed
Mix-Swin Transformer Block
$4 \times H \times W \times 3$
Feature Extraction
Bicubic Interpolation
$4 \times H \times W \times 3$
Multi-Swin Transformer Encoder T0 Block
Conv
E0
$4 \times \frac{H}{2} \times \frac{W}{2} \times 3$
Multi-Swin Transformer Encoder T1 Block
Conv
E1
$4 \times \frac{H}{4} \times \frac{W}{4} \times 3$
Multi-Swin Transformer Encoder T2 Block
Conv
E2
$4 \times \frac{H}{8} \times \frac{W}{8} \times 3$
Multi-Swin Transformer Encoder T3 Block
E3
$4 \times \frac{H}{8} \times \frac{W}{8} \times 3$
Bicubic Interpolation
$7 \times H \times W \times 3$
$7 \times H \times W \times 3$
D0
D1
D2
D3
$7 \times H \times W \times 3$
Conv
Recon-struction
$\times N$
Multi-Swin Transformer Decoder Block
$7 \times H \times W \times 3$
De-conv
Multi-Swin Transformer Decoder Block
$7 \times \frac{H}{2} \times \frac{W}{2} \times 3$
De-conv
Multi-Swin Transformer Decoder Block
$7 \times \frac{H}{4} \times \frac{W}{4} \times 3$
De-conv
Multi-Swin Transformer Decoder Block
$7 \times \frac{H}{8} \times \frac{W}{8} \times 3$
$7 \times \frac{H}{8} \times \frac{W}{8} \times 3$

Figure 2. The architecture of the proposed RVFIT. The features extracted from four input frames as processed by encoders to build dictionaries which will be utilized as inputs for the decoders . Upsample modules double the frame size by pixel shuffle. The Multi-Swin Transformer encoder and decoder block include a set of Mix Swin Transformer Blocks, which are explained in Figure 3 and 4 in detail.

3.1 Network overview

Given $n+1$ low frame rate (LFR) frames $\chi_{LFR} = \{I_{2t-1}^{LFR}\}_{t=1}^{n+1}$, a RVFIT model generates $2n+1$ high frame rate (HFR) frames $\chi_{HFR} = \{I_t^{HFR}\}_{t=1}^{2n+1}$, where t denotes the time stamp of a frame. Note that only frames with odd timestamp in χ_{HFR} has the LFR counterpart in χ_{LFR}.

Referring to [35], we designed a hierarchical U-shaped Transformer named Real-time Video Frame Interpolation Transformer (RVFIT), which spatially expands input video sequences while considering temporal fluency without dividing the model into temporal and spatial super-resolution modules. This design is superior to previous CNN-based frame interpolation methods because of its parallelism in structure, which can accelerate the inference process based on guaranteed performance.

We let f denote the basic function of the underlying logic of RVFIT, which takes 4 consecutive LFR frames in χ and outputs the same number of HFR frames in sequence:

$$f: \left(\chi_{2t-1}^{LFR}, \chi_{2t+1}^{LFR}, \chi_{2t+3}^{LFR}, \chi_{2t+5}^{LFR}\right) \mapsto \left(\chi_{2t-1}^{HFR}, \chi_{2t}^{HFR}, \chi_{2t+1}^{HFR}, \chi_{2t+2}^{HFR}, \chi_{2t+3}^{HFR}, \chi_{2t+4}^{HFR}, \chi_{2t+5}^{HFR}\right) \tag{1}$$

Figure 2 showed that RVFIT consists of four encoders $E_k, k = 0,1,2,3$ and matching decoder $D_k, k = 0,1,2,3$. In RVFIT, firstly a feature extraction block processes the four LFR frames, we denote the result as $(F_{2t-1}, F_{2t+1}, F_{2t+3}, F_{2t+5})$. Then extracted features input Multi-Swin Transformer Block τ_{swin}:

$$T_0 = \tau_{swin}(F_{2t-1}, F_{2t+1}, F_{2t+3}, F_{2t+5}) \tag{2}$$

Encoder block generated the embedded feature T_0. Φ means convolutional block from E_0 to E_2. The entire encoding process is shown below:

$$\begin{cases} T_k = \tau_{swin}(E_{k-1}), k = 1,2,3 \\ E_k = \Phi(T_k), k = 1,2 \\ E_3 = T_3 \end{cases} \tag{3}$$

Clearly, we define the four output features of each E_k:

$$E_k \equiv \left(E_{k,t}, E_{k,t+1}, E_{k,t+2}, E_{k,t+3}\right) \tag{4}$$

Actually, a reusable dictionaries [35] is built in each E_k, combining attention calculated result and relative position bias [16] of input frame windowed partitions.The detail of the encoder architecture are presented in Section 3.2.

After computing E_3, RVFIT constructs a query builder that generates features for interpolating frames at finer time stamps. We define the query Q as seven-channel feature maps with

$$\begin{aligned} Q := (&E_{3,2t-1}, \tfrac{1}{2}\left(E_{3,2t-1} + E_{3,2t+1}\right), E_{3,2t+1}, \\ &\tfrac{1}{2}\left(E_{3,2t+1} + E_{3,2t+3}\right), E_{3,2t+3}, \\ &\tfrac{1}{2}\left(E_{3,2t+3} + E_{3,2t+5}\right), E_{3,2t+5}) \end{aligned} \tag{5}$$

Now RVFIT is ready to synthesize the HFR frames by sending to the decoders with the features of different cascaded encoder outputs. The entire decoding process is shown below:

$$\begin{cases} D_3 = \Phi^{-1}(\tau_{swin}^{-1}(T_3), \\ D_k = \Phi^{-1}(\tau_{swin}^{-1}(T_k, D_{k+1})), k = 1,2 \\ D_0 = \tau_{swin}^{-1}(T_0, D_1) \end{cases} \tag{6}$$

where τ_{swin}^{-1} means the Multi-Swin Transformer Decoder Block and Φ^{-1} denotes the deconvolutional block from D_1 to D_3. We explain the details of decode architecture in Section 3.3.

For the final synthesis, RVFIT rebuilds HFR frames from residuals. In order to enhance the super-resolution effect to a certain extent, finally, we superimpose the bucubic interpolation results of each frame of the reconstructed HFR frame and LFR frame, and then pass through a layer of convolutional neural network to obtain the final HFR video sequence.

One of the key of the architecture of RVFIT is Mix Swin Transformer, which is a new architecture combine the core attention module proposed by Swin Transformer version 2 [34] and Transformer module organization of Swin Transformer version 1 [16] . The validity of this structure will be discussed in Section 6. Another key point isthe encoder-built reusable dictionaries based on LFR frames, which are used in decoders D_k to promote reconstruction of HFR frames. This advantage makes RVFIT stay ahead of many existing methods that depend on duplicate feature fusions, and thus led to an increase in computing speed and size, particularly.

3.2 Encoder

In this subsection, we introduce the encoder architecture of RVFIT. Before the data stream reaches the Transformer network, the feature extraction module has shown in Figure 2, we use one convolutional layer with kernel size 3×3 to extract C features from input RGB frames. This light-weight feature extractor is meaningfully smaller than the various residual blocks in [5] [1] .It also refines the color channel into small features to provide sufficient and detailed data for double Pixel-Shuffle upsampling [36] and transformer training.

Following the shallow feature extrator, the encoder part of RVFIT consist of four stage. $E_k, k = 0,1,2,3$ stand for each stack of Mix Swin Transformer blocks and followed convolution layer, except E_3. Through shifting non-overlapping windows to decrease computationl time consuming while keeping the power of learning long-range dependencies. Assume the size of pre-defined window is $W_{window} \times W_{window}$, a Mix Swin Transformer block divides the input frames of size $N \times H \times W_{frame} \times C$ into $N \times \lceil \frac{H}{W_{window}} \rceil \times \frac{W_{frame}}{W_{window}} \times C$ non-overlapping windows, we set $N = 2$, $W_{window} = 4$

and $C = 96$ in our all experiments. After flattining the features in each window to produce feature map of size $\frac{NHW_{frame}}{W_{window}^{2}} \times W_{window}^{2} \times C$.

Layer Normalization (LN) is employed to the features before Window-based Multi-head Self Attention (W-MSA) [16] calculates the local attention of windows. a Multi-Layer Perception (MLP) followed by an LN layer are applied to subsequent transformation. A Shifted Window-based Multi-head Self-Attention (SW-MSA) [16] is used for insert the cross-window connections. And every module is the same as the previous Swin Transformer block except that the input features are shifted by $\lfloor \frac{W_{window}}{2} \rfloor \times \lfloor \frac{W_{window}}{2} \rfloor$ before window partitioning. Eventually, the output of a set of Swin Transformer blocks are downsampled by a convolution layer with kernel size of *4* and stride of *2*, presenting as the input of both the next encoder stage and the corresponding decoder stage.

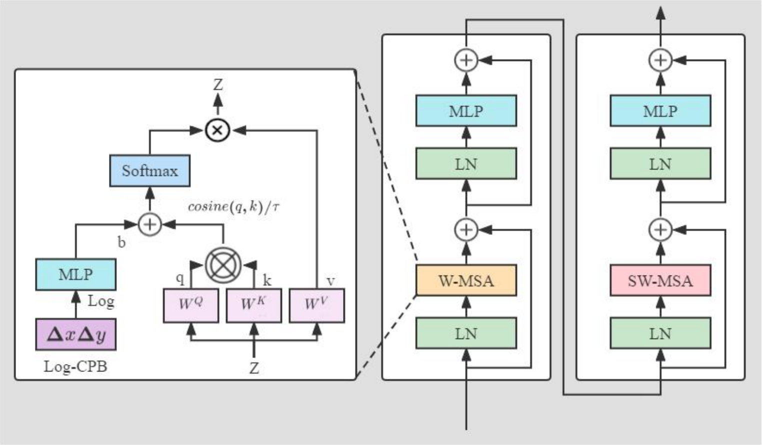

Figure 3. The Mix Swin Transformer encoder block processed . The first and second block computes multi-head self attention in each window partition and shifted window parition, respectively. LN means Layer Normalization, MLP means Multiple Layer Perception. W-MSA denotes Windowed Multi-Head Self-Attention and SW-MSA denotes Shifted Windowed Multi-Head Self-Attention.

3.3 Decoder

Same as the encoder architecture, we use four stage of decoders followed by a deconvolution layer for feature upsampling. The input of decoder $D_k, k = 0,1,2,3$ consitituted by two part: one is the batch of frame querys and the other is the output features from the encoder. In Multi Self Attention block (W-MSA and SW-MSA) the input from upper decoder layer ($D_k, k = 1,2,3$) or the output of query module plays the role of query (Q), so we call this part of input as query in below. While in the Multi Cross Attention block (W-MCA and SW-MCA) the decoder produce features of HFR-frame by parallelly and repeatly querying the dictionaries (the key-value pairs (K, V)) established from the equal level encoders E_k. Importantly, dictionaries provided by the encoders are pre-processed for reuse in corresponding blocks. In addition, the processing of data by the decoding layer can be roughly regarded as the inverse process of the encoding layer.

After the second time double Pixel-Shuffle upsampling [36], the reconstruction module processes the output features of the final decoder D_0 to generate the super-resolution video frames. An reconstruction layer is composed by two convolution layers and a ReLU [37] layer sandwichied between them. The number of reconstruction layer is optional, when the performance allows, the number of layers of B should be increased as much as possible to improve the fitting of the model.

Finally, RVFIT composites the interpolation of input frames and eventual neraul network super-resolution results then uses a convolution layer with a constant number of inbound and outbound channels to fuse the two together more appropriately.

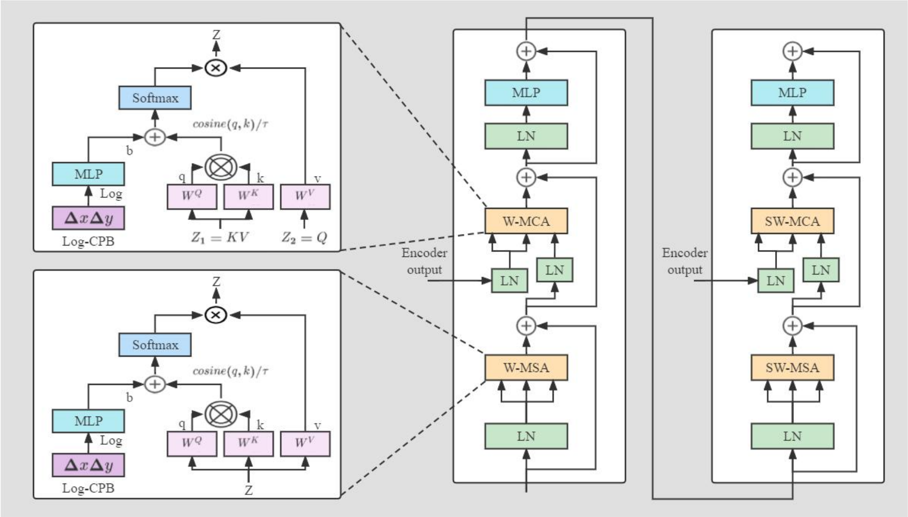

Figure 4.The Mix Swin Transformer decoder block processed $D_k, k = 0,1,2,3$. It receives a query Q and the output from the corresponding encoder E_k as the input. MCA is Multi-Head Cross Attention, and other abbreviations can refer to Figure 2.

3.4 Training Details

Optimizer. During the training of RVFIT, we select Adam as optimizer with $\mathcal{L}_2$ and decoupled weight decay [38] by setting $\beta_1 = 0.9$ and $\beta_2 = 0.99$. The initial learning rate is set to 2×10^{-4}, it is gradually decreased according to the Cosine annealing with restart [39] set to 10^{-7}. Every 30000 iteration has a restart. We use two Nvidia GeForce RTX 3090Ti and batch size set to 2, depending on the specific model architecture and training environment capability.

Objective function. The Charbonnier penalty loss [40] is applied on whole frames between the ground truth I_{GT} and restored frame interpolation frame I_{HFR}, which can be defined:

$$\ell = \sqrt{\|I_{GT} - I_{HFR}\|^2 + \epsilon^2}, \epsilon = 10^{-3} \tag{7}$$

Training and evaluating dataset. We choose Vimeo-90K [41] as our training dataset, which has over 60000 septuplet video sequences, some SOTA models also trained by it [5] [42] [1] . All the RVFIT models are evaluated on Vimeo-90K [41] and Vid4 [24] datasets. We split huge Vimeo-90K into slow, medium and fast motion sets as [5] which contains 1610, 4972 and 1225 video clips. While Vid4 is a small dataset include four video sequences. We select seven frames each time, and take four odd-numbered frames as input, then the three even-numbered frames should be the result of frame insertion, and finally compare the output result with the target of the seven frames.

4. EXPERIMENT

We design three versions of RVFIT with different scales(small (S), medium (M) and large (L)), namely RVFIT-S, RVFIT-M, RVFIT-L. In this section, we will compare them with other methods under qualitative and quantitative metrics, while discuss the advantages and limitation of RVFIT.

4.1 Quantitative evaluation

Peak Signal to Noise Ratio (PSNR) and Structural Similarity (SSIM) are evaluation metrics for quantitative test. We also present model inference rate in Frame Per Second (FPS) and model size as expressed by the number of parameters. Test results are shown in Table 1.

We analyse that all the RVSRT models achieve SOTA performance in both Vid4 and Vimeo-90K datasets with prominent smaller model size and conspicuous higher inferance rate. Additionally, the performance develops stablely with increasing number of Mix Swin Transformer blocks and reconstruction layers in the architecture, from RVSRT-S, -M to -L. Espacially, in Table 1, one can see that the smallest model RVSRT-S performs similarly as Zooming SlowMo

[22], while RVSRT-M outperforms Zooming SlowMo [22] in Vid4, Vimeo-Medium and Vimeo-Slow with significantly smaller number of parameters and faster inference speed. Our largest model RVSRT-L outperforms TMNet [1] on Vimeo-Medium, which is the largest dataset in all evaluate dataset, with 40% smaller model size. We notice that our RVSRT-S achieves a real-time rendering speed (more than 26 FPS) in keeping similar performance.

Table 1.Quantitative evaluation on several datasets with the VSR baseline methods. PSNR and SSIM based on Y channel only [22] . FPS is measured on Vid4 dataset and on Nvidia GeForce RTX 3090Ti environment. The top three best score on each test are in red, blue and black in decreasing order.

Method	Vid4		Vimeo-Fast		Vimeo-Fast		Vimeo-Fast		FPS ↑	Parameters (Millions)↓
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑		
STARNet	26.06	**0.8046**	36.19	0.9368	34.86	0.9356	33.10	**0.9164**	3.85	111.61
Zooming SlowMo	**26.31**	0.7976	**36.81**	**0.9415**	35.41	0.9361	33.36	0.9138	15.59	11.10
TMNet	**26.43**	**0.8016**	**37.04**	**0.9435**	**35.60**	**0.9380**	**33.51**	**0.9159**	14.33	12.26
RVSRT-L	**26.43**	**0.8007**	**36.79**	**0.9405**	**35.69**	**0.9381**	**33.42**	**0.9146**	**14.99**	**10.3**
RVSRT-M	**26.33**	0.7930	36.69	0.9382	**35.51**	**0.9362**	**33.39**	0.9132	**17.02**	**8.3**
RVSRT-S	26.16	0.7876	36.50	0.9371	35.46	0.9342	33.28	0.9127	**25.13**	**6.2**

4.2 Qualitative evaluation

By comparing RVFIT with other SOTA VFI methods visually in figure 5. We display three scenarios to exhibit frame interpolation effectiveness.

- The first group illustrate a video of a static calender that captured by a dynamic camera. The result of RVFIT is more vivid than others.
- The second group illustrate a video of a speeding car taken outdoors by a fixed camera. The result of RVFIT shows more texture details, such as the shape of car window and convenient red leaves.
- The third group illustrate a street full of tourists, a bird skimming the face appears in the center of the picture. How to restore the outline of the fast-moving animal and the details of the face is a very challenging task. Although RVFIT is almost as sophisticated in face generation as the counter part methods, it more appropriately restores the image of birds.

4.3 Limitations

4.3.1 lengthy training process

Like other transformer-based methods [20], the required training time of RVSRT is relatively long. It takes more than seven days for convergence with the usage of two Nvidia GeForce RTX 3090Ti cards.

4.3.2 Smooth transition without sliding window for video sequence sampling

In RVFIT, the number of video frames put into the model for processing each time is fixed. We call all frames in each processing batch as sampling sliding window. The direct scene transformation of sampling sliding window is smooth, because they are processed by the model at the same time and have continuity in time domain; However, there are some problems in the scene transformation between sampling sliding windows, such as jump, graininess and dislocation, because they do not have time-domain continuity in the processing of the model. This problem needs further research and improvement.

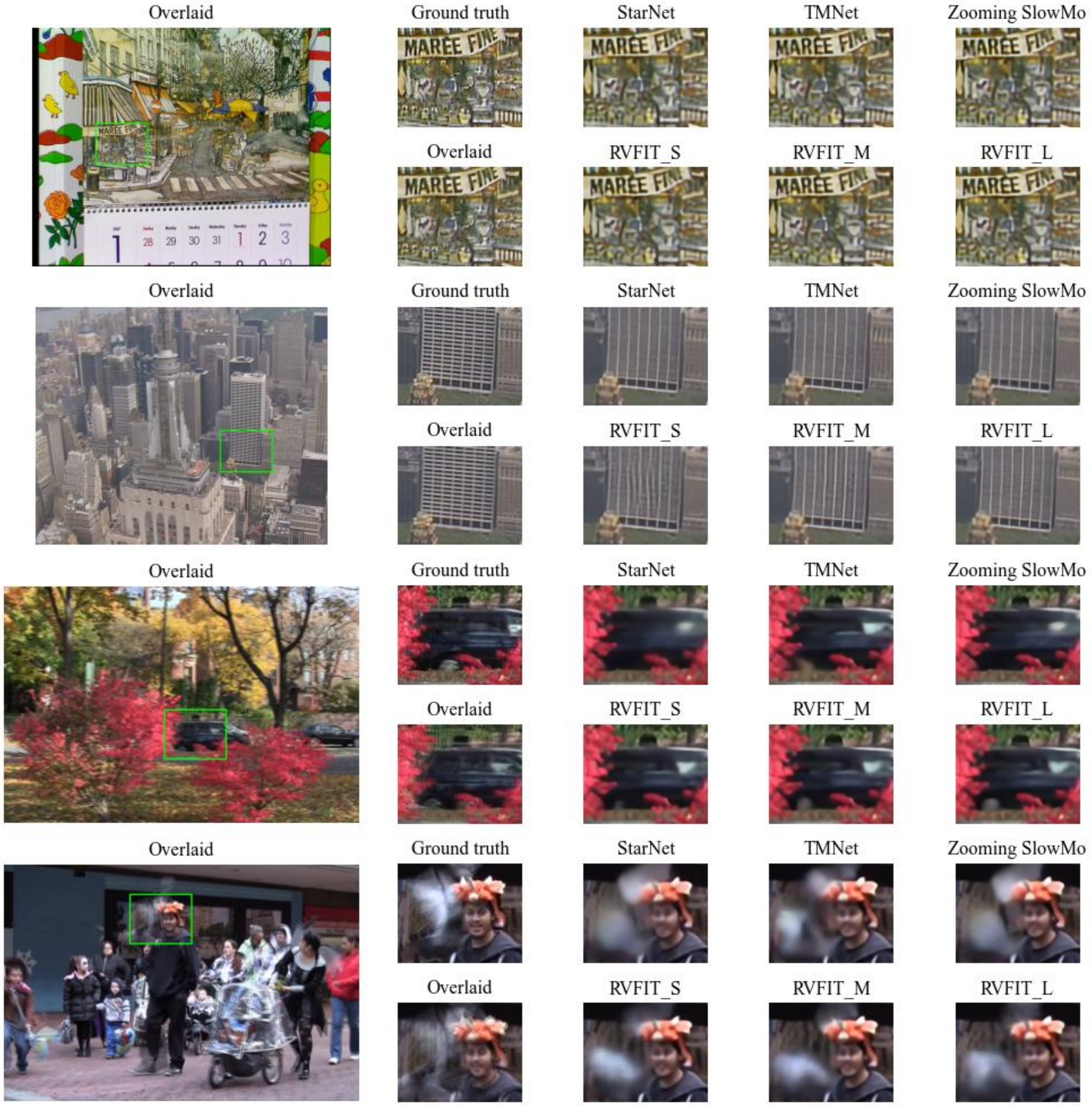

Figure 5. Comparisons on the Vid4 dataset [24]. Three RVFIT models achieve SOTA performance of visual qualities on abundant scenarios.

5. CONCLUSION

We proposed a Real-time Video Super-resolution Transformer (RVSRT) for generating HR videos from LR ones, which solves the spatial video super-resolution problem with an integrated transformer architecture that maintains temporal coherence. Particularly, LR spatial-temporal features extracted from progressive levels of encoders are used to build dictionaries, which are then queried several times in the decoding stage for interpolating HR frames simultaneously.

We emphasize that the key implementation of the work is the innovative, comprehensive formulation of the effective learning of frame features by swing transformer version 1 and the accurate relative position estimation of windowed partition by version 2. This comprehensive idea leads to an obviously lighter model with a much faster (real-time) inference speed compared with the SOTA methods without a significant difference in model performance.

The future development direction of this work includes but is not limited to: an improved super-resolution method for spatial, for example, combining Motion Estimation (ME) and Motion Compensation (MC); reusable dictionaries constructed in different levels of encoders promote computational efficiency; exploring the possibility of time-domain super-resolution; and more advanced training optimizer and objective functions that helps to enhance training capability.

REFERENCES.

[1] G. Xu, J. Xu, Z. Li, L. Wang, X. Sun, and M.-M. Cheng, "Temporal modulation network for controllable space-time video super-resolution," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 6388–6397.

[2] C.-Y. Wu, N. Singhal, and P. Krahenbuhl, "Video compression through image interpolation," in Proceedings of the european conference on computer vision (ECCV), 2018, pp. 416 - 431.

[3] M. Haris, G. Shakhnarovich, and N. Ukita, "Space-time-aware multi-resolution video enhancement," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2859–2868.

[4] S. Y. Kim, J. Oh, and M. Kim, "Fisr: Deep joint frame interpolation and super-resolution with a multi-scale temporal loss," in *Proceedings of the AAAI conference on artificial intelligence*, 2020, vol. 34, pp. 11278–11286.

[5] X. Xiang, Y. Tian, Y. Zhang, Y. Fu, J. P. Allebach, and C. Xu, "Zooming slow-mo: Fast and accurate one-stage space-time video super-resolution," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 3370–3379.

[6] J. Flynn, I. Neulander, J. Philbin, and N. Snavely, "Deepstereo: Learning to predict new views from the world's imagery," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5515–5524.

[7] N. K. Kalantari, T.-C. Wang, and R. Ramamoorthi, "Learning-based view synthesis for light field cameras," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, pp. 1–10, 2016.

[8] W. Bao, W.-S. Lai, C. Ma, X. Zhang, Z. Gao, and M.-H. Yang, "Depth-aware video frame interpolation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3703–3712.

[9] H. Jiang, D. Sun, V. Jampani, M.-H. Yang, E. Learned-Miller, and J. 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*, 2018, pp. 9000–9008.

[10] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, and A. Agarwala, "Video frame synthesis using deep voxel flow," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4463–4471.

[11] S. Niklaus and F. Liu, "Context-aware synthesis for video frame interpolation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1701–1710.

[12] S. Niklaus, L. Mai, and F. Liu, "Video frame interpolation via adaptive separable convolution," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 261–270.

[13] T. Peleg, P. Szekely, D. Sabo, and O. Sendik, "Im-net for high resolution video frame interpolation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2398–2407.

[14] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*, 2020, pp. 213–229.

[15] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[16] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[17] T. Brown *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[19] A. Vaswani *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[20] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International conference on medical image computing and computer-assisted intervention*, 2015, pp. 234–241.

[21] X. Chu *et al.*, “Twins: Revisiting the design of spatial attention in vision transformers,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 9355–9366, 2021.

[22] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, “Swinir: Image restoration using swin transformer,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 1833–1844.

[23] Z. Wang, X. Cun, J. Bao, W. Zhou, J. Liu, and H. Li, “Uformer: A general u-shaped transformer for image restoration,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 17683–17693.

[24] C. Liu and D. Sun, “A bayesian approach to adaptive video super resolution,” in *CVPR 2011*, 2011, pp. 209–216.

[25] B. Tag, J. Shimizu, C. Zhang, K. Kunze, N. Ohta, and K. Sugiura, “In the eye of the beholder: The impact of frame rate on human eye blink,” in *Proceedings of the 2016 CHI conference extended abstracts on human factors in computing systems*, 2016, pp. 2321–2327.\

[26] H. Lee, T. Kim, T. Chung, D. Pak, Y. Ban, and S. Lee, “Adacof: Adaptive collaboration of flows for video frame interpolation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 5316–5325.

[27] S. Niklaus, L. Mai, and O. Wang, “Revisiting adaptive convolutions for video frame interpolation,” in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2021, pp. 1099–1109.

[28] J. Park, K. Ko, C. Lee, and C.-S. Kim, “Bmbc: Bilateral motion estimation with bilateral cost volume for video interpolation,” in *European conference on computer vision*, 2020, pp. 109–125.

[29] H. Sim, J. Oh, and M. Kim, “Xvfi: Extreme video frame interpolation,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 14489–14498.

[30] X. Xu, J. Pan, Y.-J. Zhang, and M.-H. Yang, “Motion blur kernel estimation via deep learning,” *IEEE Transactions on Image Processing*, vol. 27, no. 1, pp. 194–205, 2017.

[31] X. Xu, J. Pan, Y.-J. Zhang, and M.-H. Yang, “Motion blur kernel estimation via deep learning,” *IEEE Transactions on Image Processing*, vol. 27, no. 1, pp. 194–205, 2017.

[32] X. Xu and C. C. Loy, “3D human texture estimation from a single image with transformers,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 13849–13858.

[33] F. Yang, H. Yang, J. Fu, H. Lu, and B. Guo, “Learning texture transformer network for image super-resolution,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 5791–5800.

[34] Z. Liu *et al.*, “Swin transformer V2: Scaling up capacity and resolution,” *arXiv preprint arXiv:2111.09883*, 2021.

[35] Z. Geng, L. Liang, T. Ding, and I. Zharkov, “RSTT: Real-time spatial temporal transformer for space-time video super-resolution,” *arXiv preprint arXiv:2203.14186*, 2022.

[36] W. Shi *et al.*, “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1874–1883.

[37] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 315–323.

[38] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” *arXiv preprint arXiv:1711.05101*, 2017.

[39] A. Gotmare, N. S. Keskar, C. Xiong, and R. Socher, “A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation,” *arXiv preprint arXiv:1810.13243*, 2018.

[40] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, “Deep laplacian pyramid networks for fast and accurate super-resolution,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 624–632.

[41] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, “Video enhancement with task-oriented flow,” *International Journal of Computer Vision*, vol. 127, no. 8, pp. 1106–1125, 2019.

[42] X. Xiang, Y. Tian, Y. Zhang, Y. Fu, J. P. Allebach, and C. Xu, “Zooming slowmo: An efficient one-stage framework for space-time video super-resolution,” *arXiv preprint arXiv:2104.07473*, 2021.

Authors' Information

Your Name	**Title***	**Research Field**	**Homepage**
First Author	master student	Audio/Video AI Enhancement, Audio/Video Coding	
Second Author	full professor	Key technology research, data intelligence applications for large information systems	