RVSRT: Real-time Video Super Resolution 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 super-resolution is the task of converting low-resolution video to high-resolution video. Existing methods with better intuitive effects are mainly based on convolutional neural networks (CNNs), but the architecture is heavy, resulting in a slow inference structure. Aiming at this problem, this paper proposes a real-time video super-resolution.

Real-time video super resolution transformer (RVSRT) can quickly complete the super-resolution task while considering the visual fluency of video frame switching. Unlike traditional methods based on CNNs, this paper does not process video frames separately with different network modules in the temporal 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 (6.1M vs 12.3M, parameters) while ensuring comparable performance, and the speed is increased by 80% (26.2 fps vs 14.3 fps, frame size is 720*576).

Keywords: Video super resolution, vision transformer, deep learning.

1. INTRODUCTION

Video super-resolution (VSR) aims to reconstruct high-resolution video (HR) from low-resolution (LR) video. As one of the fundamental problems in the field of computer vision, VSR is often used to improve visual quality [2], and has excellent value in many practical applications, such as video surveillance [3], HDTV [4], and satellite imagery [5] [6]. Regarding methodological differences, unlike image super-resolution learning from the spatial domain, the VSR task focuses more on exploring correlations in the temporal domain. Video quality can be significantly improved if the detailed texture used to restore the target frame can be found and exploited in adjacent frames.

In order to meet this challenge, researchers have successfully proposed many VSR methods in recent years, which can be roughly divided into two categories. One class exploits temporal information through a recurrent mechanism [7] [8] [9]. One of the representative works is Icon-VSR, which uses a hidden state to transfer relevant features in the entire frame [7]. Recurrent networks often lack long-term modeling capabilities due to vanishing gradients [10], leading to unsatisfactory results. Another class attempts to utilize adjacent frames as input (e.g. frames 5 to 7) and align temporal features either implicitly [11] [12] or explicitly [13] [2]. One of the classic works is EDVR [2], which employs deformable convolutions to capture features within a sliding window. However, a larger window size will significantly increase the computational cost, making this paradigm incapable of capturing distant frames.

Inspired by recent advances in natural language processing [16] by Transformers, significant progress has been made in both productive tasks [17] [18] and visual recognition [19] [20]. For example, MuCAN proposes to use an attention mechanism to aggregate inter-frame features for VSR tasks [21]. Nevertheless, due to the high computational complexity of the video, it can only learn from a narrow time window, which leads to sub-optimal performance. Therefore, exploring the correct way to use Transformers in videos remains a considerable challenge.

In this paper, we propose a novel real-time video super-resolution Transformer to enable effective video reconstruction learning (RVSRT). This approach leads to a much lighter-weight network compared with the previous

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

methods and is able to a real-time inference speed without sacrificing much performance as Figure 1. Specifically, our contributions as summarized as follows:

- We propose a Real-time Video Super-resolution Transformer (RVSRT) to combine traditional interpolation algorithms and neural networks, this resolution without explicitly modeling it as two separate tasks. To the best of our realize, it is the first time that a Transformer takes advantage of traditional algorithms.

- Inside RVSRT, 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 RVSRT models with architectures with different complexity, resulting in small (S), medium (M), and large (L) architectures. Experiments show that RVSRT is remarkably faster and smaller than the SOTA VSR methods while maintaining similar performance

  - RVSRT-L performs similarly to TMNet [1] with 18% fewer parameters, RVSRT-M outperforms Zooming SlowMo [22] with 8% fewer parameters, and RVSRT-S outperforms STARNet [23] with 45% fewer parameters.

  - RVSRT-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 [22] with a 75% speedup and outperforms STARNet [23] with around 700% speedup.

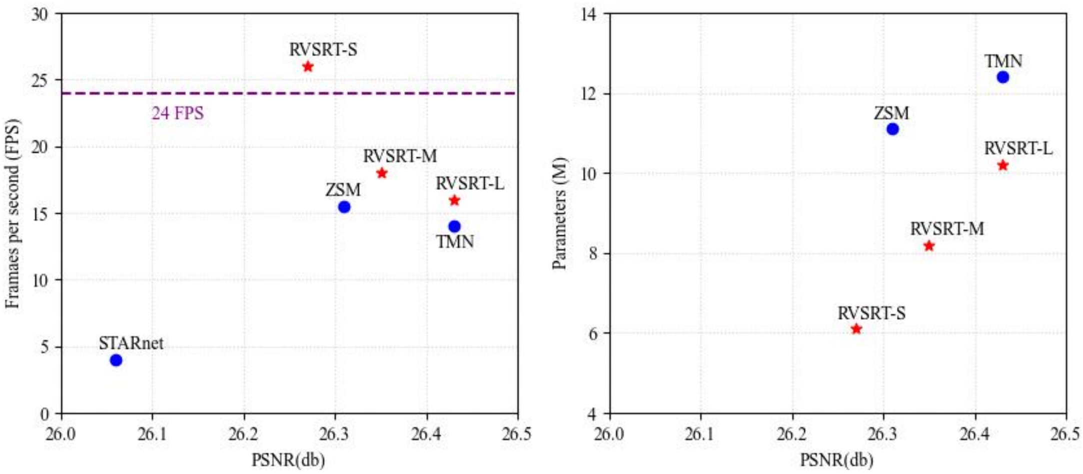

Figure 1. Performance of RVSRT on Vid4 dataset [14] using small (S), medium (M) and large (L) architectures compared to other baseline models. The upper sub-figure displays FPS versus PSNR. Espacially 24 FPS is the standard cinematic frame rate [15] . The lower sub-figure presents parameter number (in millions) versus PSNR.

2. RELATED WORK

2.1 Video super-resolution (VSR)

In VSR tasks, it is essential to support frame reconstruction with other frames in the sequence. Therefore, according to the length of the input video sequence, VSR methods can probably be divided into two paradigms: based recurrent structure and sliding-window structure.

Methods based on recurrent structure use a hidden state to transport pertinent information in previous frames. RSDN [24] divided the input into structure and detail components and used the two-steam structure-detail block to learn textures. RBPN [25] regarded each frame as a separate source, integrated into an iterative subtlety framework. FRVSR [26] utilized the previously super-resolution frame to recover the subsequent frame. Typically, IconVSR [7], OVSR [9], and BasicVSR [7] merged the bi-directional hidden state from the past and future for recovery and got marked improvements. They attempt to completely use the information of the entire sequence and synchronously update the hidden state by the weights of the reconstruction network. Nevertheless, due to the vanishing gradient [10], this mechanism makes the updated hidden state loses its long-term modeling abilities to some degree.

Rather than aggregating information from enduring hidden states, the methods based on sliding-window structure use adjacent frames within in a sliding window as inputs to synthesize the HR frame. They primarily concentrate on using 2D or 3D CNN [24] [27] [11] [12], deformable convolutions [28] [13] [2] or optical flow estimation [29] [30] [31] to design excellent alignment modules and combine detailed textures form near frames. TDAN [13] effectively introduces deformable convolution to align temporal features firmly and achieves impressive performance, while EDVR [2] incorporates deformable convolution into a multi-scale module to improve the feature alignments to a greater extent. FSTRN [12] offered a fast spatial-temporal residual network for VSR by selecting 3D convolutions [32] to utilize the additional information across frames. VESCPN [29] proposed a spatial-temporal sub-pixel convolution network and first combined the motion compensation and VSR. However, they cannot employ textures at other moments, especially in relatively distant frames.

2.2Video Transformer

Transformer [16] is an overall architecture in Natural Language Processing (NLP) and achieves SOTA performance in various tasks [33] [34] . Lately, Transformers have become popular in CV field. The advanced Vision Transformer (ViT) [20] computes attention between flattened image patches to solve image classification tasks and exceeds CNN-based methods. TTSR [17] introduced a texture Transformer in low-level vision to find relevant texture patches from refer image to LR image. To solve VSR problems, VSR-Transformer [35] and MuCAN [21] tried to choose attention mechanisms for aligning different frames with great success. However, because of the high computational costs of attention calculation on videos, these methods only aggregate information on the narrow temporal window. Liu et al. [36] [37] 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. Afterward, [38] designed a U-shape network based on Swin Transformer for general image restoration. SwinIR [39] using Swin Transformer to handle the image restoration task and proposed residual Swin Transformer blocks. RSTT [40] built a spatial-temporal transformer that naturally incorporates the spatial and temporal super-resolution modules into a single model.

In this paper, we introduced light and fast real-time VSR network based on Swin Transformer. Instead of constructing dictionaries and queries from identical single frames, we used them to compute window and shifted window attentions from multiple frames simultaneously. This characteristic results reduction of model size and accelerated inference.

3. THE PROPOSED APPROACH

In this section, we first give an overview of RVSRT 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

3.1 Network overview

Given n LR frames $\chi^L = \{I_t^L\}_t^n$ of size $H \times W \times 3$, a RVSRT model generates n HR frames $\chi^H = \{I_t^H\}_t^n$ of size $4H \times 4W \times 3$, where t denotes the time stamp of a frame. Each HR frame in χ^H has the LR counterpart χ^L.

Referring to [40], we designed a hierarchical U-shaped Transformer named Real-time Video Super-Resolution Transformer (RVSRT), which spatially expands LR video sequences while taking into account temporal fluency without dividing the model into temporal and spatial super-resolution modules. This design is superior to previous CNN-based super-resolution methods because of its parallelism in structure, which can accelerate the inference process on the basis of guaranteed performance.

We let f denote the basic function of the underlying logic of RVSRT, which takes 4 consecutive LR frames in χ^L and outputs the same number of HR frames in sequence:

$$f: \left(\chi_t^L, \chi_{t+1}^L, \chi_{t+2}^L, \chi_{t+3}^L\right) \mapsto \left(\chi_t^L, \chi_{t+1}^H, \chi_{t+2}^H, \chi_{t+3}^H\right) \tag{1}$$

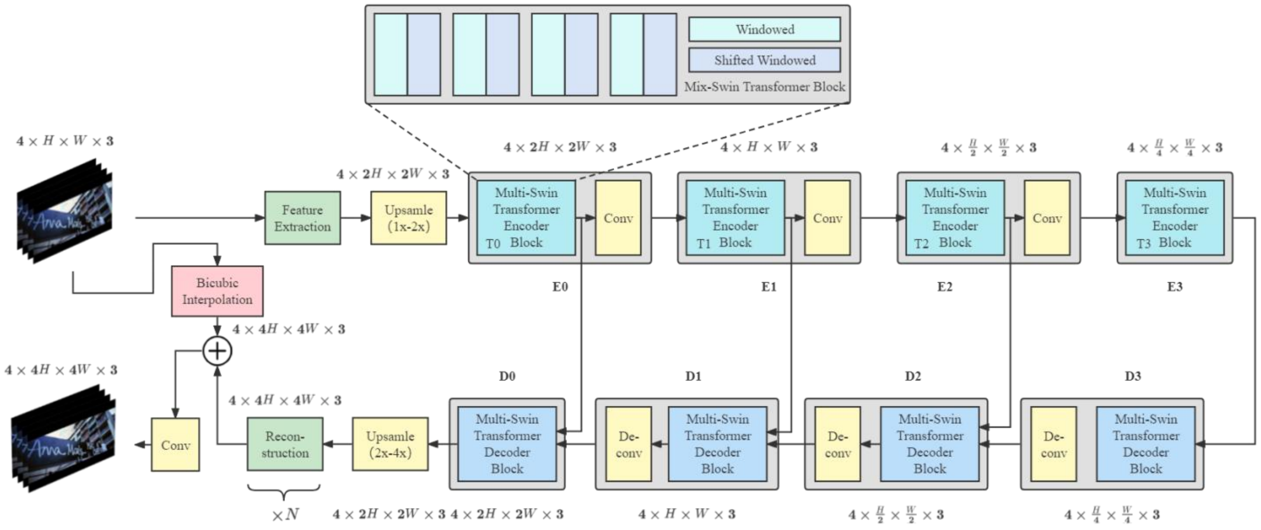
Windowed
Shifted Windowed
Mix-Swin Transformer Block
$4 \times H \times W \times 3$
$4 \times 2H \times 2W \times 3$
Feature Extraction
Upsamle (1x-2x)
Multi-Swin Transformer Encoder T0 Block
Conv
E0
$4 \times H \times W \times 3$
Multi-Swin Transformer Encoder T1 Block
Conv
E1
$4 \times \frac{H}{2} \times \frac{W}{2} \times 3$
Multi-Swin Transformer Encoder T2 Block
Conv
E2
$4 \times \frac{H}{4} \times \frac{W}{4} \times 3$
Multi-Swin Transformer Encoder T3 Block
E3
Bicubic Interpolation
$4 \times 4H \times 4W \times 3$
$4 \times 4H \times 4W \times 3$
$4 \times 4H \times 4W \times 3$
Conv
Recon-struction
$\times N$
Upsamle (2x-4x)
$4 \times 2H \times 2W \times 3$
D0
Multi-Swin Transformer Decoder Block
$4 \times 2H \times 2W \times 3$
D1
De-conv
Multi-Swin Transformer Decoder Block
$4 \times H \times W \times 3$
D2
De-conv
Multi-Swin Transformer Decoder Block
$4 \times \frac{H}{2} \times \frac{W}{2} \times 3$
D3
De-conv
Multi-Swin Transformer Decoder Block
$4 \times \frac{H}{4} \times \frac{W}{4} \times 3$

Figure 2. The architecture of the proposed RVSRT. The features extracted from four LR frames as processed by encoders $E_k, k = 0,1,2,3$ to build dictionaries which will be utilized as inputs for the decoders $D_k, k = 0,1,2,3$. 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.

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

$$T_0 = \tau_{swin}\left(F^L_t, F^L_{t+1}, F^L_{t+2}, F^L_{t+3}\right) \tag{2}$$

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 [40] is built in each E_k, combining attention calculated result and relative position bias [36] of LR frame windowed partitions.The detail of the encoder architecture are presented in Section 3.2.

After computing E_3, RVSRT is now ready to synthesize the HR frames by sending to the decoders with the features of different cascaded encoder outputs. The entire decoding process is shown below:

$$\begin{cases} D_3 = \Phi^{-1}(\tau^{-1}_{swin}(T_3), \\ D_k = \Phi^{-1}(\tau^{-1}_{swin}(T_k, D_{k+1})), k = 1,2 \\ D_0 = \tau^{-1}_{swin}(T_0, D_1) \end{cases} \tag{5}$$

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, RVSRT rebuilds HR 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 HR frame and LR frame, and then pass through a layer of convolutional neural network to obtain the final HR video sequence.

One of the key of the architecture of RVSRT is Mix Swin Transformer, which is a new architecture combine the core attention module proposed by Swin Transformer version 2 [37] and Transformer module organization of Swin Transformer version 1 [36] . The validity of this structure will be discussed in Section 6. Another key point isthe encoder-built reusable dictionaries based on LR frames, which are used in decoders D_k to promote reconstruction of HR frames. This advantage makes RVSRT 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 RVSRT. 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 LR RGB frames. This light-weight feature extractor is meaningfully smaller than the various residual blocks in [22] [41] [1] .It also refines the color channel into small features to provide sufficient and detailed data for double Pixel-Shuffle upsampling [42] and transformer training.

Following the shallow feature extrator, the encoder part of RVSRT 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 $\frac{NHW_{frame}}{W_{window}^2} \times W_{window}^2 \times C$.

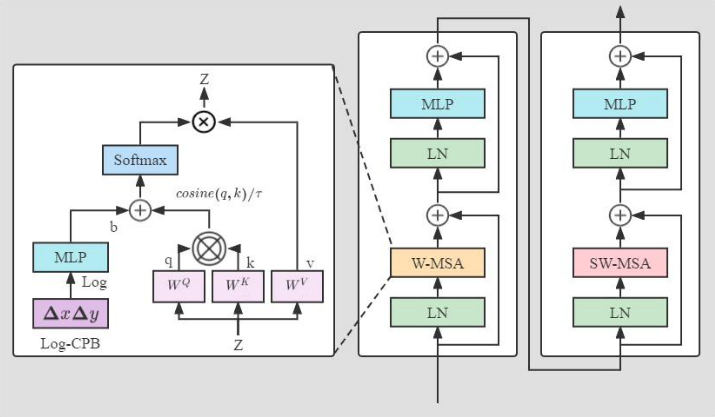

Figure 3. The Mix Swin Transformer encoder block processed . The first and second block computes multi-head self attention in each window partition and shifted window parition, respectively. LN means Layer Normalization, MLP means Multiple Layer Perception. W-MSA denotes Windowed Multi-Head Self-Attention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

Layer Normalization (LN) is employed to the features before Window-based Multi-head Self Attention (W-MSA) [36] 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) [36] 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.

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 last encoder layer (D_0) 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 output-frame by parallelly and repeaty 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 [42], 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 [43] 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, RVSRT composites the interpolation of LR 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.

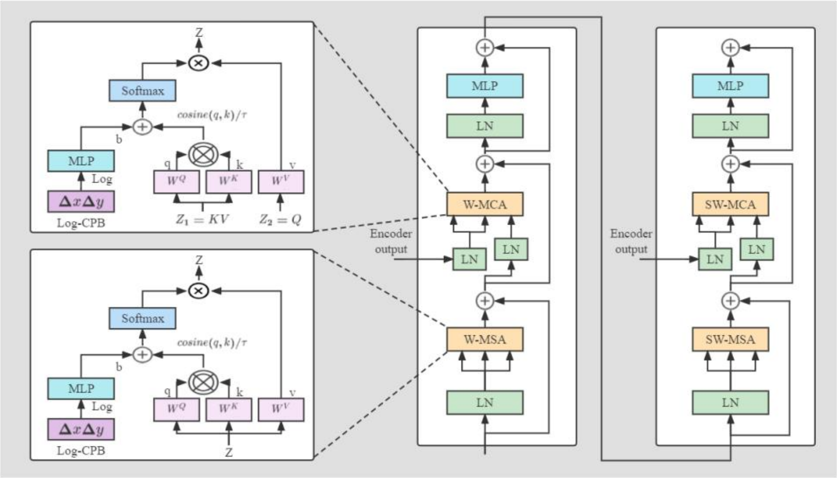

Figure 4. The Mix Swin Transformer decoder block processed $D_k, k = 0,1,2,3$. It receives a query Q and the output from the corresponding encoder E_k as the input. MCA is Multi-Head Cross Attention, and other abbreviations can refer to Figure 2.

3.4 Training Details

3.4.1 Optimizer

During the training of RVSRT, we select Adam as optimizer with $\mathcal{L}_2$ and decoupled weight decay [44] 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 [45] 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.

3.4.2 Objective function

The Charbonnier penalty loss [46] is applied on whole frames between the ground truth I_{HR} and restored super-resolution frame I_{SR}, which can be defined:

$$\ell = \sqrt{\|I_{HR} - I_{SR}\|^2 + \epsilon^2}, \epsilon = 10^{-3} \tag{6}$$

3.4.3 Training and evaluating dataset

We choose Vimeo-90K [47] as our training dataset, which has over 60000 septuplet video sequences, some SOTA models also trained by it [22] [41] [1] . The size of input LR frame is 112 × 164 and length of a single input frames set is four. In the case of four times super resolution, the size of HR frames is 448 × 226.

All the RVSRT models are evaluated on Vimeo-90K [47] and Vid4 [14] datasets. We split huge Vimeo-90K into slow, medium and fast motion sets as [22] which contains 1610, 4972 and 1225 video clips. Vid4 is a small dataset include four video sequences with 180 × 144 input size and 720 × 576 output size.

4. EXPERIMENTS

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

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**	**0.8012**	**36.82**	**0.9413**	**35.71**	**0.9382**	**33.50**	**0.9150**	**14.99**	**10.2**
RVSRT-M	**26.35**	0.7996	36.72	0.9402	**35.63**	**0.9364**	**33.48**	0.9145	**19.86**	**8.2**
RVSRT-S	26.27	0.7941	36.53	0.9381	35.49	0.9351	33.35	0.9131	**26.19**	**6.1**

4.2 Qualitative evaluation

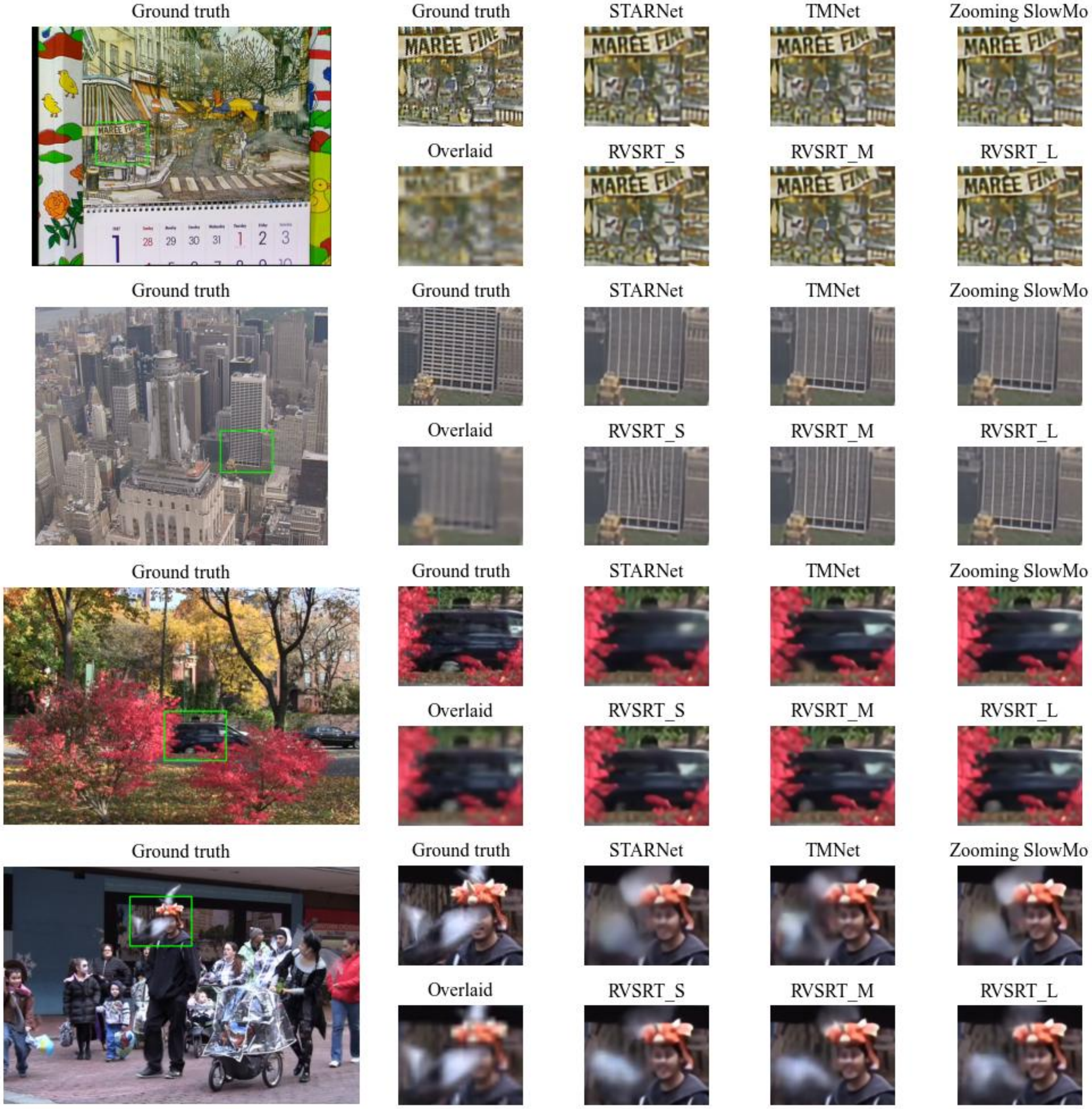

Figure 5. Visual comparisons on Vid4 dataset [14]. RVSRT with three different sizes of architectures outperform the baseline methods.

By comparing RVSRT with other SOTA VSR methods visually in figure 5. We display three scenarios to exhibit super-resolution effectiveness.

- The first and second group illustrate a video of a static calender that captured by a dynamic camera. The result of RVSRT is more vivid than others.
- The third group illustrate a video of a speeding car taken outdoors by a fixed camera. The result of RVSRT shows more texture details, such as the shape of car window and convenient red leaves.

- The forth 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 RVSRT 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 RVSRT, 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.

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] X. Wang, K. C. Chan, K. Yu, C. Dong, and C. Change Loy, “Edvr: Video restoration with enhanced deformable convolutional networks,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, 2019, pp. 0–0.

[3] L. Zhang, H. Zhang, H. Shen, and P. Li, “A super-resolution reconstruction algorithm for surveillance images,” Signal Processing, vol. 90, no. 3, pp. 848–859, 2010.

[4] T. Goto, T. Fukuoka, F. Nagashima, S. Hirano, and M. Sakurai, “Super-resolution system for 4K-HDTV,” in 2014 22nd international conference on pattern recognition, 2014, pp. 4453–4458.

[5] M. Deudon et al., “Highres-net: Recursive fusion for multi-frame super-resolution of satellite imagery,” arXiv preprint arXiv:2002.06460, 2020.

[6] Y. Luo, L. Zhou, S. Wang, and Z. Wang, “Video satellite imagery super resolution via convolutional neural networks,” IEEE Geoscience and Remote Sensing Letters, vol. 14, no. 12, pp. 2398–2402, 2017.

[7] K. C. Chan, X. Wang, K. Yu, C. Dong, and C. C. Loy, “BasicVSR: The search for essential components in video super-resolution and beyond,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 4947–4956.

[8] R. Vemulapalli, M. Brown, and S. M. M. Sajjadi, “Frame-recurrent video super-resolution.” Google Patents, 2020.

[9] P. Yi et al., “Omniscient video super-resolution,” in Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 4429–4438.

[10] S. Hochreiter, “The vanishing gradient problem during learning recurrent neural nets and problem solutions,” International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 6, no. 2, pp. 107–116, 1998.

[11] S. Y. Kim, J. Lim, T. Na, and M. Kim, “3dsrnet: Video super-resolution using 3d convolutional neural networks,” arXiv preprint arXiv:1812.09079, 2018.

[12] S. Li, F. He, B. Du, L. Zhang, Y. Xu, and D. Tao, “Fast spatio-temporal residual network for video super-resolution,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 10522–10531.

[13] Y. Tian, Y. Zhang, Y. Fu, and C. Xu, “Tdan: Temporally-deformable alignment network for video super-resolution,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 3360–3369.

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

[15] 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.

[16] A. Vaswani et al., “Attention is all you need,” Advances in neural information processing systems, vol. 30, 2017.

[17] 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.

[18] Y. Zeng, J. Fu, and H. Chao, “Learning joint spatial-temporal transformations for video inpainting,” in European conference on computer vision, 2020, pp. 528–543.

[19] 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.

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

[21] W. Li, X. Tao, T. Guo, L. Qi, J. Lu, and J. Jia, “Mucan: Multi-correspondence aggregation network for video super-resolution,” in European conference on computer vision, 2020, pp. 335–351.

[22] 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.

[23] 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.

[24] T. Isobe, X. Jia, S. Gu, S. Li, S. Wang, and Q. Tian, “Video super-resolution with recurrent structure-detail network,” in European conference on computer vision, 2020, pp. 645–660.

[25] M. Haris, G. Shakhnarovich, and N. Ukita, “Recurrent back-projection network for video super-resolution,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 3897–3906.

[26] M. S. Sajjadi, R. Vemulapalli, and M. Brown, “Frame-recurrent video super-resolution,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 6626–6634.

[27] Y. Jo, S. W. Oh, J. Kang, and S. J. Kim, “Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 3224–3232.

[28] J. Dai et al., “Deformable convolutional networks,” in Proceedings of the IEEE international conference on computer vision, 2017, pp. 764 – 773.

[29] J. Caballero et al., “Real-time video super-resolution with spatio-temporal networks and motion compensation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4778 – 4787.

[30] T. H. Kim, M. S. Sajjadi, M. Hirsch, and B. Scholkopf, “Spatio-temporal transformer network for video restoration,” in Proceedings of the european conference on computer vision (ECCV), 2018, pp. 106 – 122.

[31] X. Tao, H. Gao, R. Liao, J. Wang, and J. Jia, “Detail-revealing deep video super-resolution,” in Proceedings of the IEEE international conference on computer vision, 2017, pp. 4472 – 4480.

[32] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3d convolutional networks,” in Proceedings of the IEEE international conference on computer vision, 2015, pp. 4489 – 4497.

[33] 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.

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

[35] J. Cao, Y. Li, K. Zhang, and L. Van Gool, “Video super-resolution transformer,” arXiv preprint arXiv:2106.06847, 2021.

[36] Z. Liu et al., “Swin transformer: Hierarchical vision transformer using shifted windows,” in Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 10012 – 10022.

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

[38] Z. Wang, X. Cun, J. Bao, and J. Liu, “Uformer: A general u-shaped transformer for image restoration,” arXiv preprint arXiv:2106.03106, 2021.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

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

[45] 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.

[46] 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.

[47] 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.

Authors' background

Your Name	Title*	Research Field	Personal website
Linlin Ou	master student	Audio/Video AI Enhancement, Audio/Video Coding	
Yuanping Chen	full professor	Key technology research, data intelligence applications for large information systems	