Transformer Design for Low-Level Vision Tasks

Huan Yang

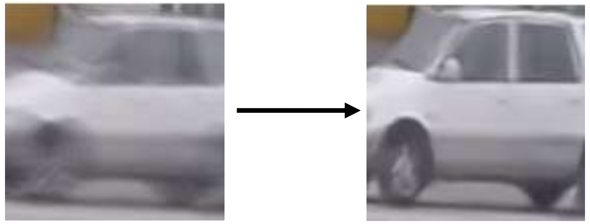
Microsoft Research Asia

Transformer in Low-Level Vision

Denoise

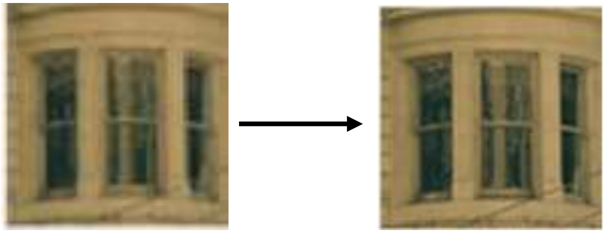
SUNet, CSformer

Deblur



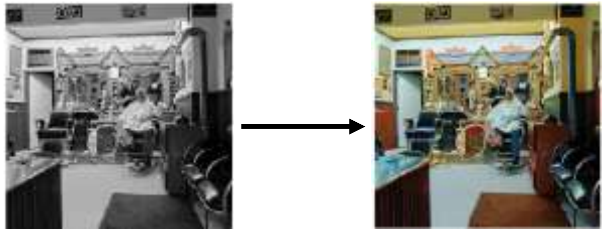
Stripformer

Super-Resolution



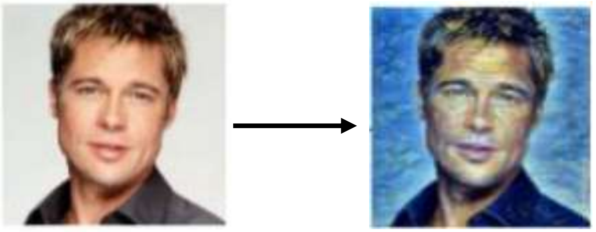
TTSR, ESRT, HAT

Colorization



ColTran

Stylization



StyTr2, STTR

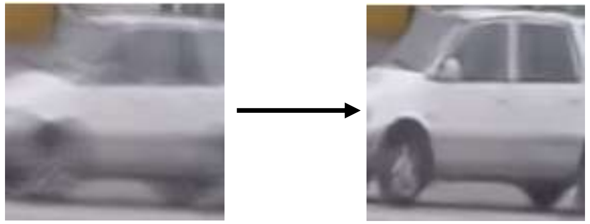
Restoration,
Enhancement,
...

Transformer in Low-Level Vision

Denoise

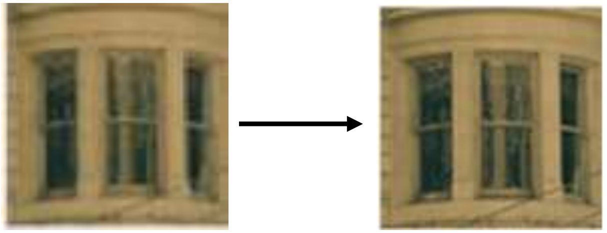
SUNet, CSformer

Deblur



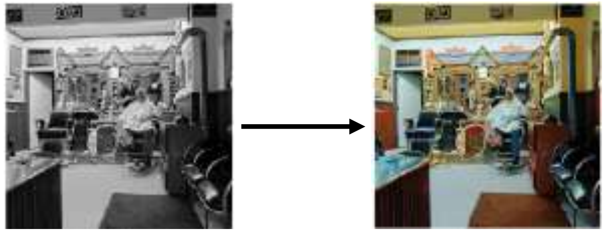
Stripformer

Super-Resolution



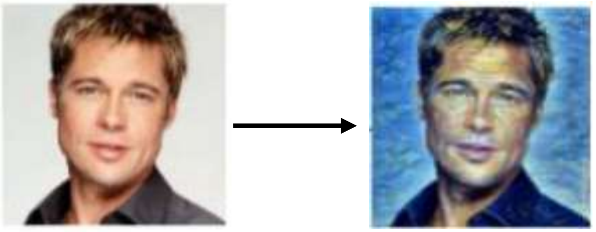
TTSR, ESRT, HAT

Colorization



ColTran

Stylization



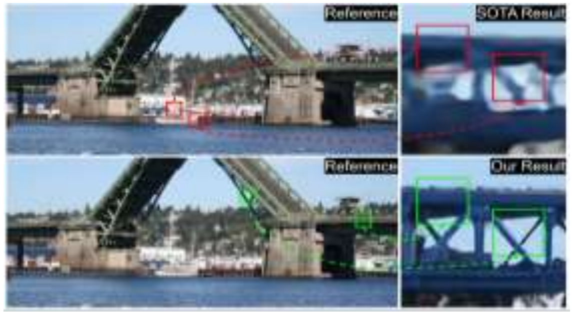
StyTr2, STTR

Restoration,
Enhancement,
...

Transformer for Super-Resolution

CVPR 2020, TTSR

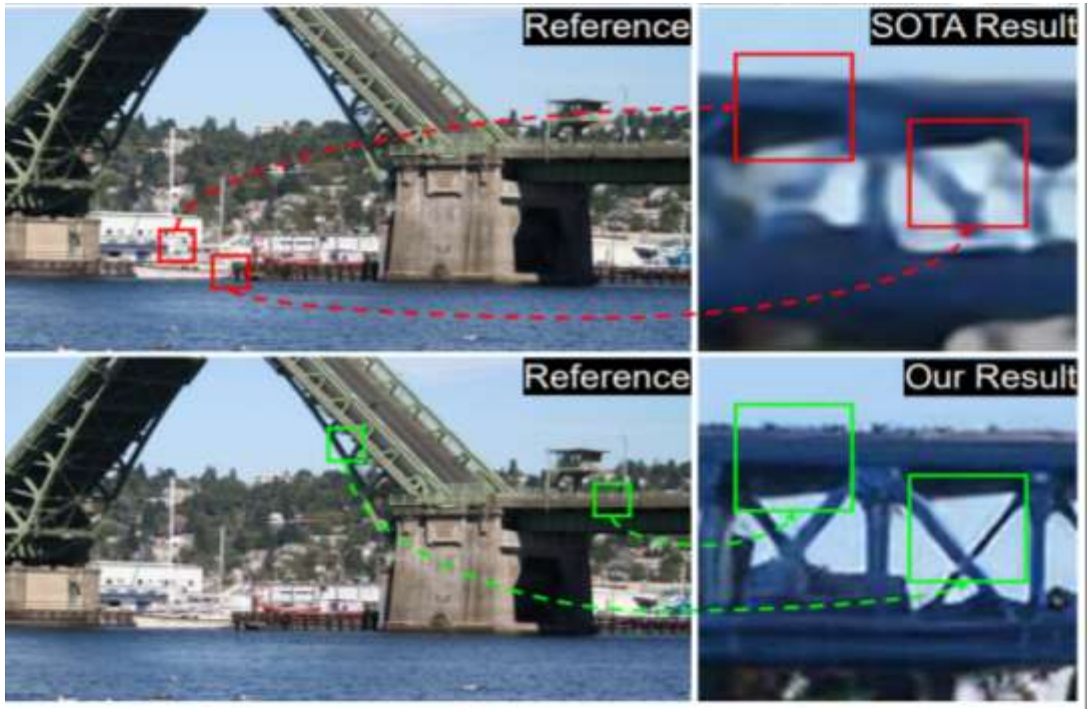
Texture Transformer
for Image SR



+0.85db
SRNTT

Challenges in Image SR

- The high-frequency texture is hard to be recovered from the LR image itself. Even with some powerful networks and adversarial training, the results still suffer from unrealistic artifacts.
- Introducing external high-quality images as references has shown great potential for image super-resolution.



Internal Texture Recovery → External Texture Transfer

Texture-Transformer for RefSR

- Texture Search: learn a joint texture feature embedding for image super-resolution

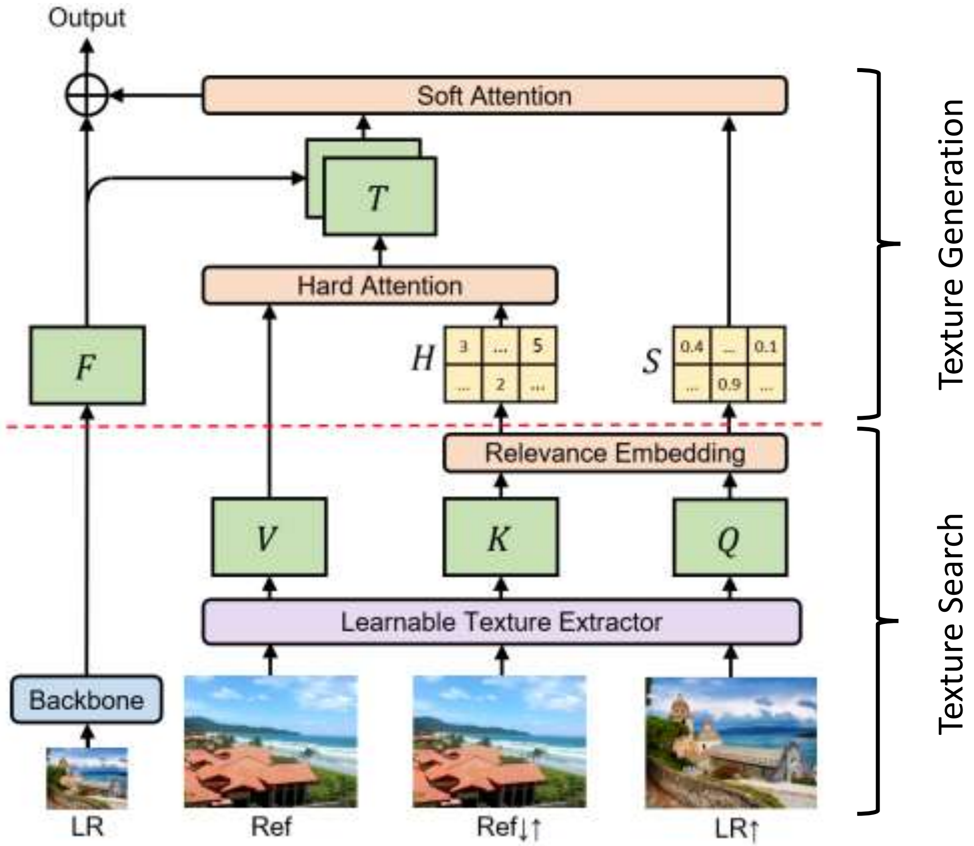
$$f_{LR} = \phi_{LTE_\theta}(LR), f_{Ref} = \phi_{LTE_\theta}(Ref)$$

$$r_{i,j} = \prod_l \left\langle \frac{f^i_{LR_l}}{|f^i_{LR_l}|}, \frac{f^j_{Ref_l}}{|f^j_{Ref_l}|} \right\rangle$$

- Texture Generation: encourage relevant texture transfer and avoid wrong texture through a multi-attentional generator

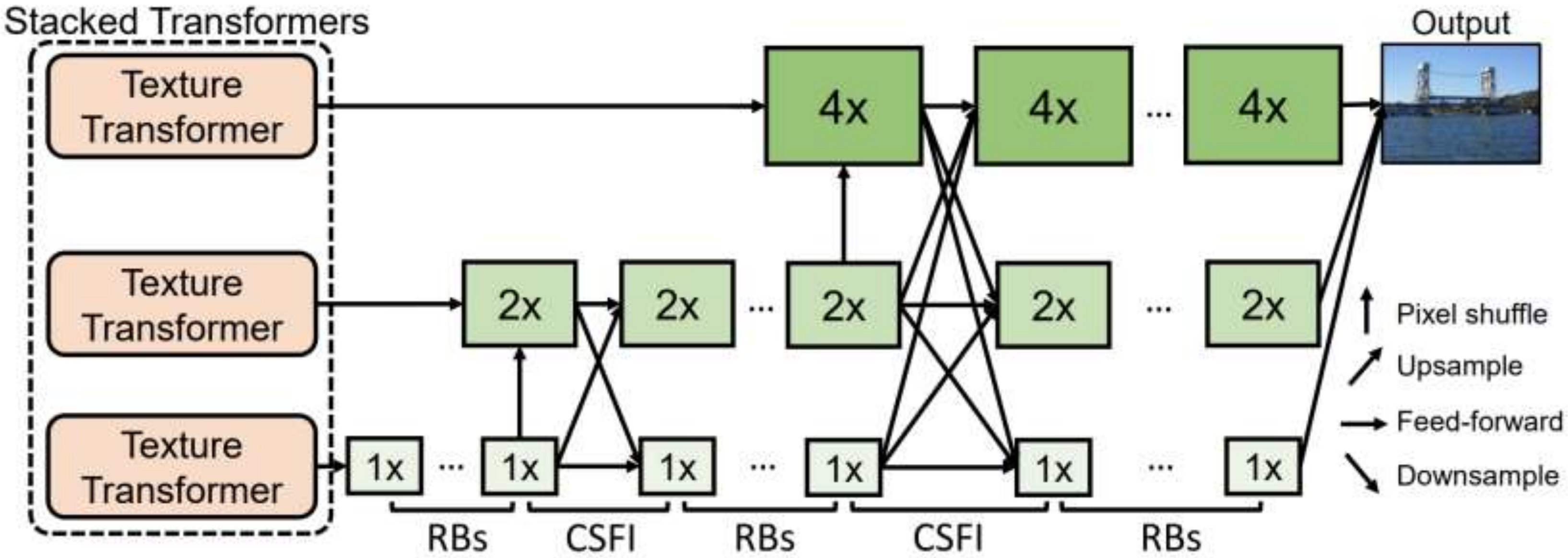
$$h_i = \arg\max_j r_{i,j}, \; s_i = \max_j r_{i,j}, \; f^i_T = f^{h_i}_{Ref}$$

$$f = f + Conv(Concate(f, f_T)) \odot S$$



Texture-Transformer for RefSR

- We proposed a cross-scale feature integration (CSFI) module to stack more Transformer blocks.

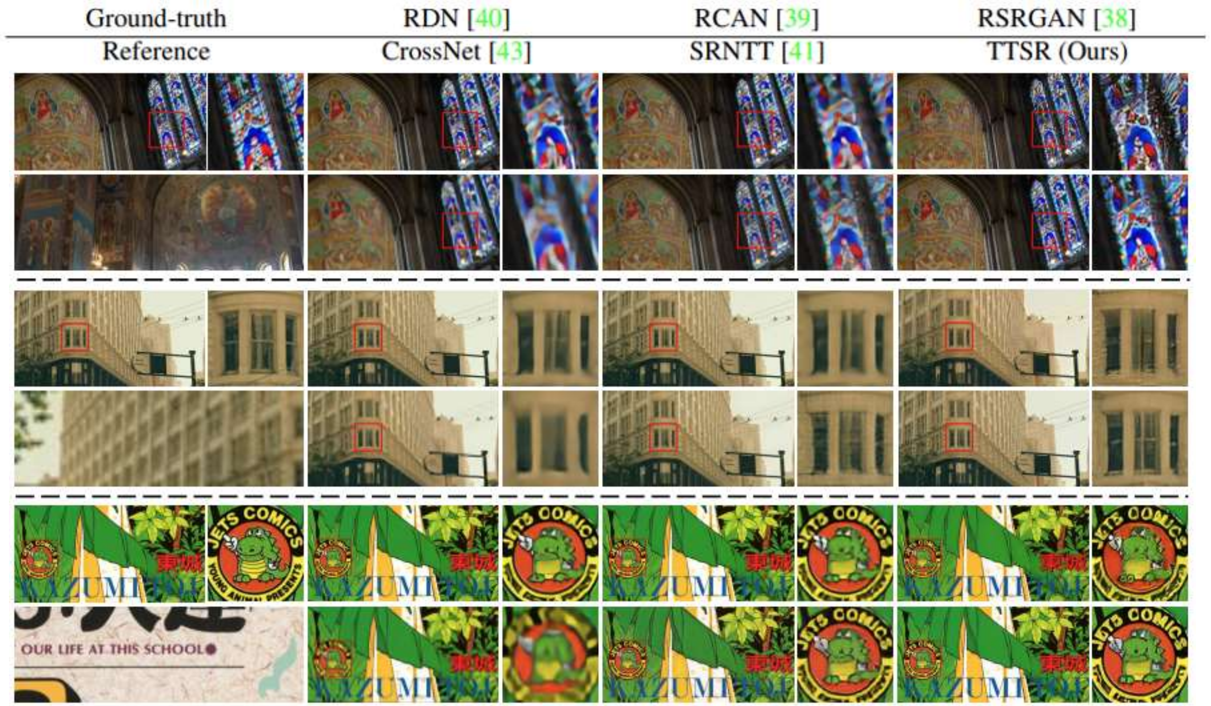


Performance

- Our method significant outperforms existing SISR and RefSR methods by a large margin.

Method	CUFED5	Sun80	Urban100	Manga109
SRCNN [3]	25.33 / .745	28.26 / .781	24.41 / .738	27.12 / .850
MDSR [17]	25.93 / .777	28.52 / .792	25.51 / .783	28.93 / .891
RDN [40]	25.95 / .769	29.63 / .806	25.38 / .768	29.24 / .894
RCAN [39]	26.06 / .769	29.86 / .810	25.42 / .768	29.38 / .895
SRGAN [16]	24.40 / .702	26.76 / .725	24.07 / .729	25.12 / .802
ENet [22]	24.24 / .695	26.24 / .702	23.63 / .711	25.25 / .802
ESRGAN [32]	21.90 / .633	24.18 / .651	20.91 / .620	23.53 / .797
RSRGAN [38]	22.31 / .635	25.60 / .667	21.47 / .624	25.04 / .803
CrossNet [43]	25.48 / .764	28.52 / .793	25.11 / .764	23.36 / .741
SRNTT-*rec* [41]	26.24 / .784	28.54 / .793	25.50 / .783	28.95 / .885
SRNTT [41]	25.61 / .764	27.59 / .756	25.09 / .774	27.54 / .862
TTSR-*rec*	27.09 / .804	30.02 / .814	25.87 / .784	30.09 / .907
TTSR	25.53 / .765	28.59 / .774	24.62 / .747	28.70 / .886

Visual Results



More Results

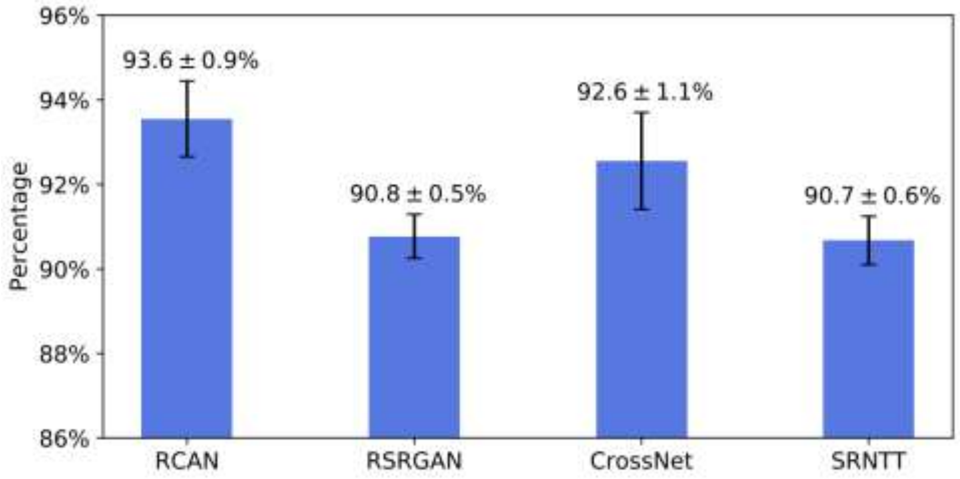
- Ablation for Texture Transformer

Method	HA	SA	LTE	PSNR/SSIM
Base				26.34 / .780
Base+HA	✓			26.59 / .786
Base+HA+SA	✓	✓		26.81 / .795
Base+HA+SA+LTE	✓	✓	✓	26.92 / .797

- Ablation for Cross-Scale Learning

Method	CSFI	numC	param.	PSNR/SSIM
Base+TT		64	4.42M	26.92 / .797
Base+TT+CSFI	✓	64	6.42M	27.09 / .804
Base+TT(C80)		80	6.53M	26.93 / .797
Base+TT(C96)		96	9.10M	26.98 / .799

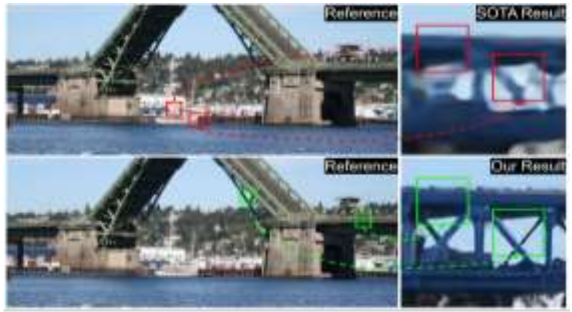
- User Study



Transformer for Super-Resolution

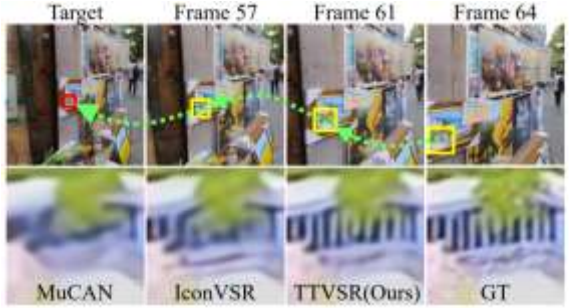
CVPR 2020, TTSR

Texture Transformer
for Image SR



CVPR 2022 Oral, TTVSR

Trajectory Transformer
for Video SR

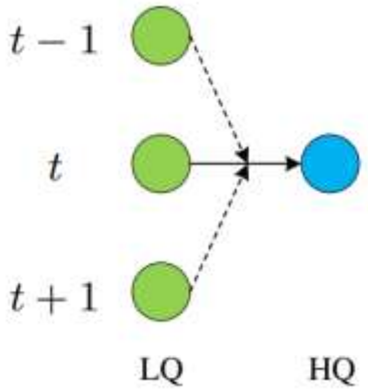


+0.85db
SRNTT

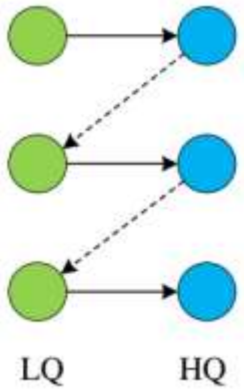
+0.70db
BasicVSR

Challenges in Video SR

- The core challenge in VSR is how to leverage temporal information. **Sliding window-based** methods aggregate adjacent 3 or 5 frames for the reconstruction. **Recurrent based** methods maintain a hidden state for temporal information.
- Is it possible to have a more efficient way to directly leverage **long-range (> 10 frames)** temporal information?



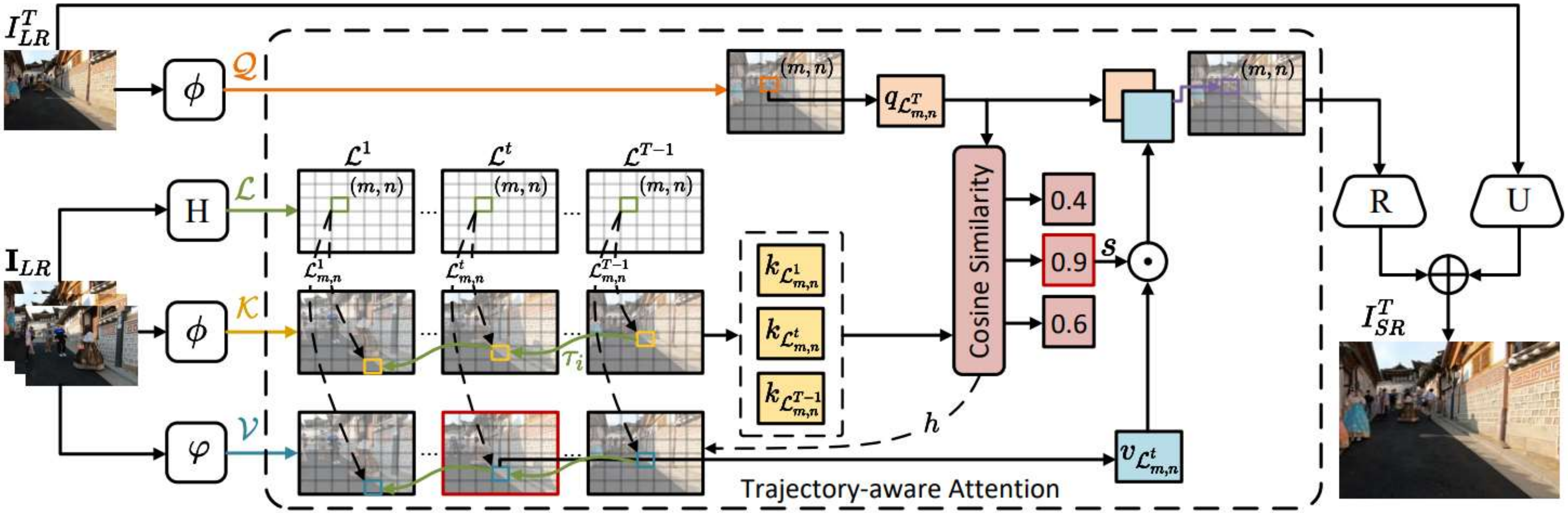
Sliding window



Recurrent

Trajectory-Aware Transformer for VSR

- We are the first that propose to leverage temporal information of visual tokens **only along its motion trajectory** inside a Transformer.



Trajectory-Aware Attention

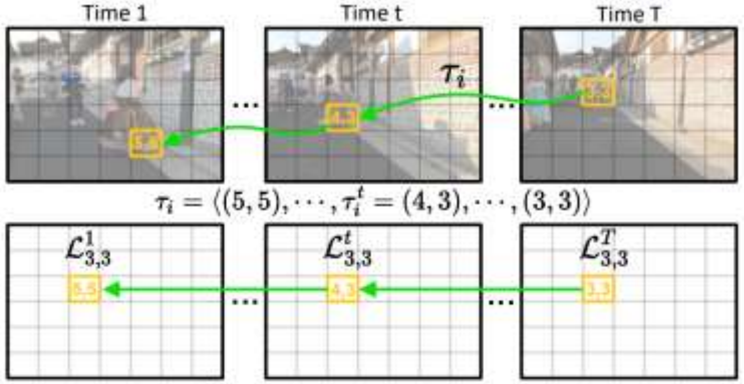
- Attention Mechanism

$$h_{\tau_i} = \arg\max_t \langle \frac{q_{\tau_i^T}}{\| q_{\tau_i^T} \|_2^2}, \frac{k_{\tau_i^t}}{\| k_{\tau_i^t} \|_2^2} \rangle,$$

$$s_{\tau_i} = \max_t \langle \frac{q_{\tau_i^T}}{\| q_{\tau_i^T} \|_2^2}, \frac{k_{\tau_i^t}}{\| k_{\tau_i^t} \|_2^2} \rangle.$$

$$\mathrm{A}_{traj}(q_{\tau_i^T}, k_{\tau_i}, v_{\tau_i}) = \mathrm{C}(q_{\tau_i^T}\ ,\ s_{\tau_i} \odot v_{\tau_i^{h_{\tau_i}}})$$

- Location Map for Trajectory Generation



$$\mathcal{L}_{m,n}^t = \tau_i^t, \text{ where } \tau_i^T = (m,n),\ i \in [1, N],$$

$$^*\mathcal{L}^t = \mathrm{S}(\mathcal{L}^t, O^{T+1}),$$

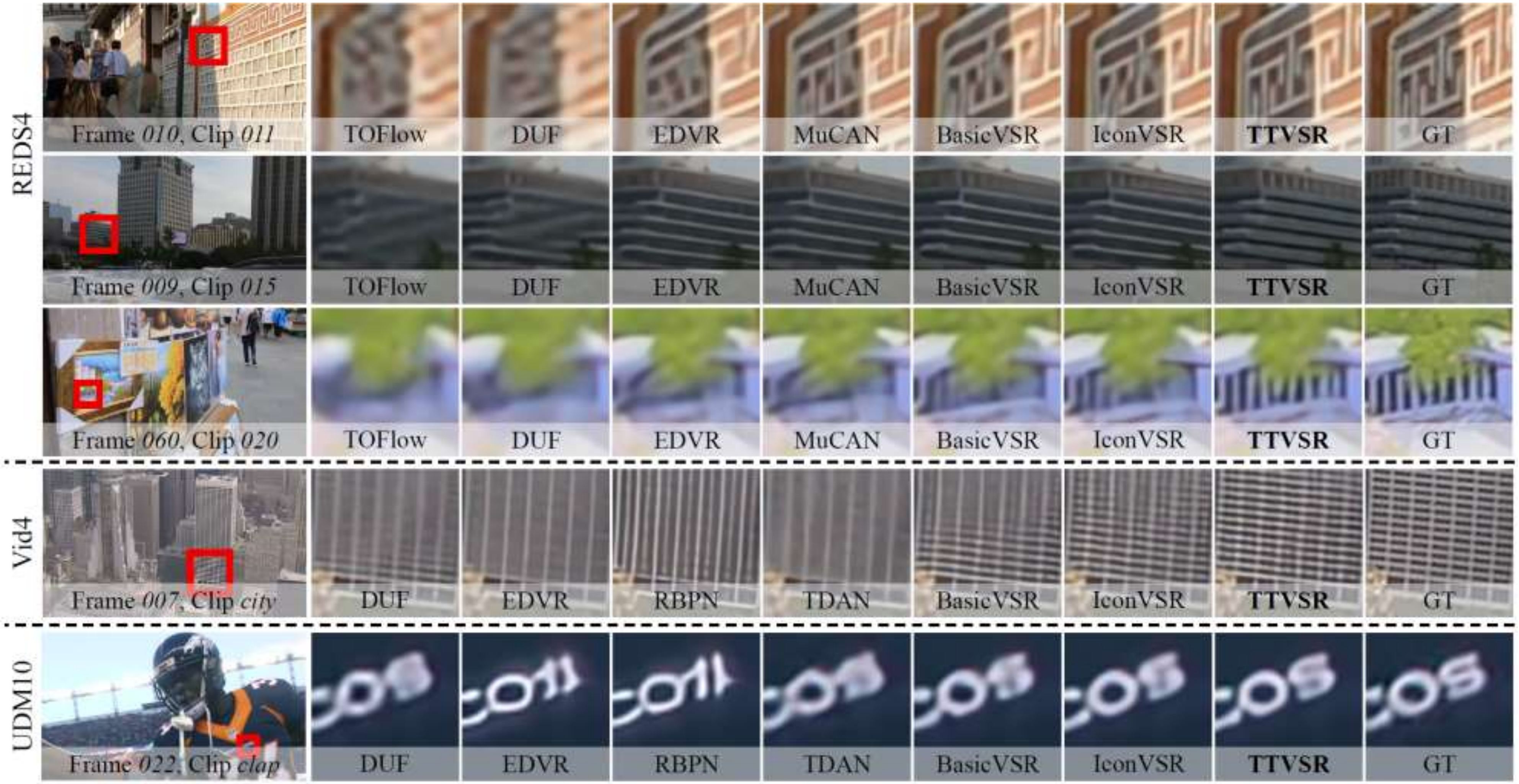
Reduce the computational cost from $(T \cdot \frac{H}{D_h} \cdot \frac{W}{D_w}) \cdot (C \cdot D_h \cdot D_w)$ to $(T \cdot 1 \cdot 1) \cdot (C \cdot D_h \cdot D_w)$

Performance

- Our proposed TTVSR significantly outperforms the latest SOTA BasicVSR/IconVSR by 0.70db/0.45db.

Method	#Frame	Clip_000	Clip_011	Clip_015	Clip_020	Average
Bicubic	1	24.55/0.6489	26.06/0.7261	28.52/0.8034	25.41/0.7386	26.14/0.7292
RCAN [51]	1	26.17/0.7371	29.34/0.8255	31.85/0.8881	27.74/0.8293	28.78/0.8200
CSNLN [29]	1	26.17/0.7379	29.46/0.8260	32.00/0.8890	27.69/0.8253	28.83/0.8196
TOFlow [43]	7	26.52/0.7540	27.80/0.7858	30.67/0.8609	26.92/0.7953	27.98/0.7990
DUF [17]	7	27.30/0.7937	28.38/0.8056	31.55/0.8846	27.30/0.8164	28.63/0.8251
EDVR [40]	7	28.01/0.8250	32.17/0.8864	34.06/0.9206	30.09/0.8881	31.09/0.8800
MuCAN [24]	5	27.99/0.8219	31.84/0.8801	33.90/0.9170	29.78/0.8811	30.88/0.8750
VSR-T [2]	5	28.06/0.8267	32.28/0.8883	34.15/0.9199	30.26/0.8912	31.19/0.8815
BasicVSR [4]	r	28.39/0.8429	32.46/0.8975	34.22/0.9237	30.60/0.8996	31.42/0.8909
IconVSR [4]	r	28.55/0.8478	32.89/0.9024	34.54/0.9270	30.80/0.9033	31.67/0.8948
TTVSR	r	28.82/0.8566	33.47/0.9100	35.01/0.9325	31.17/0.9094	32.12/0.9021

Static Visual Results



Dynamic Visual Results

More Results

- Ablation for Frame Number

#Frame	5	10	20	33	45
PSNR	31.89	31.93	31.97	31.99	32.01
SSIM	0.8984	0.8994	0.9005	0.9007	0.9004

- Params, FLOPs, and Latency

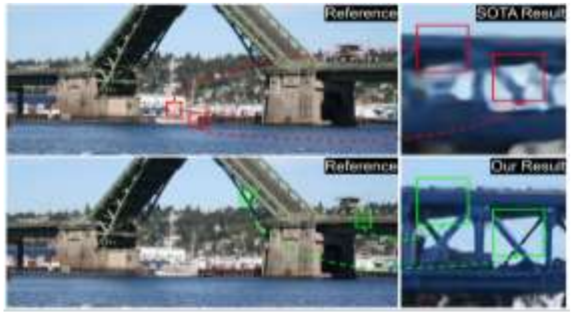
Method	#Params(M)	FLOPs(T)	PSNR/SSIM
DUF [17]	5.8	2.34	28.63/0.8251
RBPN [11]	12.2	8.51	30.09/0.8590
EDVR [40]	20.6	2.95	31.09/0.8800
MuCAN [24]	13.6	>1.07	30.88/0.8750
BasicVSR [4]	6.3	0.33	31.42/0.8909
IconVSR [4]	8.7	0.51	31.67/0.8948
TTVSR	6.8	0.61	32.12/0.9021

Method	#Params	Runtime
Flow Estimator	1.4M	11ms
Feature Extraction	0.4M	3ms
Cross-scale Feature Tokenization	0.0M	8ms
Trajectory-aware Attention	0.1M	114ms
Reconstruction Network	4.8M	72ms
TTVSR Total	6.7M	203ms
MuCAN [24]	13.6M	1,202ms
BasicVSR [4]	6.3M	63ms
IconVSR [4]	8.7M	70ms

Transformer for Super-Resolution

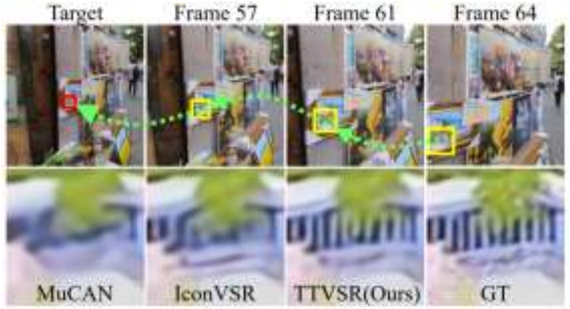
CVPR 2020, TTSR

Texture Transformer
for Image SR



CVPR 2022 Oral, TTVSR

Trajectory Transformer
for Video SR



ECCV 2022, FTVSR

Frequency Transformer
for Compressed Video SR

+0.85db
SRNTT

+0.70db
BasicVSR

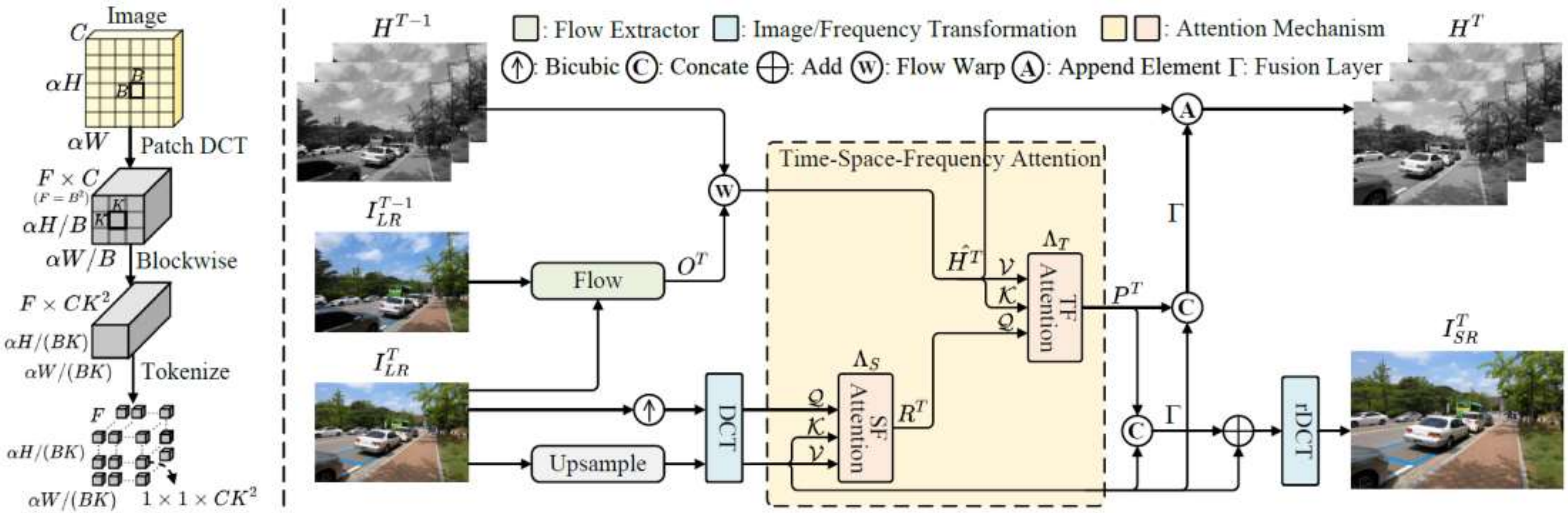
+1.58db
COMISR

Challenges in Compressed Video SR

- It is hard for existing VSR methods to leverage temporal information since it has been significantly **damaged during the compression process**.
- Considering most of such damages happen in the quantization process in some **specific frequency band** in the frequency domain.
- Is it possible to directly learn in the frequency domain to **effectively distinguish content and compression artifacts** from compressed video frames?

Frequency Transformer for CVSR

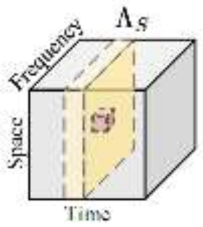
- We are the first that propose to learn super-resolution on compressed videos in the **frequency domain** and further study the attention mechanism between **spatial-temporal-frequency** dimensions.

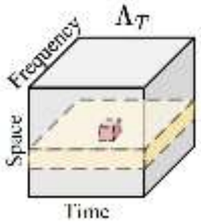


Spatial-Temporal-Frequency Attention

$$\mathcal{Q} = \{\tau^q_{(T,i,f)}, i \in [1,N], f \in [1,F]\},$$
$$\mathcal{K} = \{\tau^k_{(t,i,f)}, t \in [1,T-1], i \in [1,N], f \in [1,F]\},$$
$$\mathcal{V} = \{\tau^v_{(t,i,f)}, t \in [1,T-1], i \in [1,N], f \in [1,F]\},$$

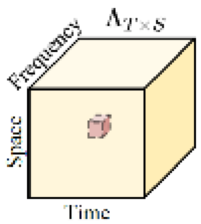
- Space/Time-Frequency Attention





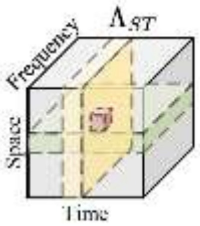
$$\Lambda_S(\tau^q_{(i,f)}, \tau^k_{(\hat{i},\hat{f})}, \tau^v_{(\hat{i},\hat{f})}), \hat{i} \in [1,N], \hat{f} \in [1,F]$$
$$\Lambda_T(\tau^q_{(t,f)}, \tau^k_{(\hat{t},\hat{f})}, \tau^v_{(\hat{t},\hat{f})}), \hat{t} \in [1,T-1], \hat{f} \in [1,F]$$

- Space-Time-Frequency Attention



$$\Lambda_{T\times S}(\tau^q_{(t,i,f)}, \tau^k_{(\hat{t},\hat{i},\hat{f})}, \tau^v_{(\hat{t},\hat{i},\hat{f})})$$

$$\Lambda_{TS}(\tau^q_{(t,i,f)}, \tau^k_{(\hat{t},\hat{i},\hat{f})}, \tau^v_{(\hat{t},\hat{i},\hat{f})})$$



$$\Lambda_{ST}(\tau^q_{(t,i,f)}, \tau^k_{(\hat{t},\hat{i},\hat{f})}, \tau^v_{(\hat{t},\hat{i},\hat{f})}) = \Lambda_T(\hat{\tau}^q_{(t,f)}, \tau_{(\hat{t},\hat{f})}, \tau_{(\hat{t},\hat{f})}),$$
$$where\ \hat{\tau} = \Lambda_S(\tau^q_{(i,f)}, \tau^k_{(\hat{i},\hat{f})}, \tau^k_{(\hat{i},f)}), \hat{t} \in [1,T-1], \hat{i} \in [1,N], \hat{f} \in [1,F]$$

Performance

- Our proposed FTVSR significantly outperforms the latest SOTA COMISR by 1.58db on CRF25.

Method	Per clip with Compression CRF25				Average of clips with Compression		
	Clip_000	Clip_011	Clip_015	Clip_020	CRF15	CRF25	CRF35
DUF [13]	23.46/0.622	24.02/0.686	25.76/0.773	23.54/0.689	25.61/0.775	24.19/0.692	22.17/0.588
FRVSR [27]	24.25/0.631	25.65/0.687	28.17/0.770	24.79/0.694	27.61/0.784	25.72/0.696	23.22/0.579
EDVR [30]	24.38/0.629	26.01/0.702	28.30/0.783	25.21/0.708	28.72/0.805	25.98/0.706	23.36/0.600
TecoGan [3]	24.01/0.624	25.39/0.682	27.95/0.768	24.48/0.686	26.93/0.768	25.46/0.690	22.95/0.589
RSDN [12]	24.04/0.602	25.40/0.673	27.93/0.766	24.54/0.676	27.66/0.768	25.48/0.679	23.03/0.579
MuCAN [18]	24.39/0.628	26.02/0.702	28.25/0.781	25.17/0.707	28.67/0.804	25.96/0.705	23.55/0.600
BasicVSR [2]	24.37/0.628	26.01/0.702	28.13/0.777	25.21/0.709	29.05/0.814	25.93/0.704	23.22/0.596
IconVSR [2]	24.35/0.627	26.00/0.702	28.16/0.777	25.22/0.709	29.10/0.816	25.93/0.704	23.22/0.596
COMISR [20]	24.76/0.660	26.54/0.722	29.14/0.805	25.44/0.724	28.40/0.809	26.47/0.728	23.56/0.599
FTVSR	26.06/0.703	28.71/0.779	30.17/0.839	27.26/0.782	30.51/0.853	28.05/0.776	24.82/0.657

Static Visual Results

Dynamic Visual Results

More Results

- Ablation for Transformer vs CNN

Domain + Backbone	Per clip with Compression CRF25				Average of clips with Compression		
	Clip_000	Clip_011	Clip_015	Clip_020	CRF15	CRF25	CRF35
Pixel + CNN	24.37/0.628	26.01/0.702	28.13/0.777	25.21/0.709	29.05/0.814	25.93/0.704	23.22/0.596
Frequency + CNN	24.98/0.666	27.11/0.746	29.36/0.818	26.05/0.751	29.20/0.825	26.87/0.745	23.83/0.629
Frequency + Transformer	25.20/0.684	27.53/0.763	29.47/0.828	26.33/0.766	29.51/0.837	27.15/0.759	24.03/0.644
Frequency + FTVSR	25.26/0.609	27.75/0.766	29.62/0.831	26.47/0.772	29.70/0.843	27.28/0.763	24.22/0.646

- Ablation for Attention Mechanism

Attention	Base	Λ_S	Λ_T	$\Lambda_{T\times S}$	Λ_{TS}	Λ_{ST}
CRF15	29.51/0.837	29.63/0.840	29.60/0.840	29.61/0.839	29.65/0.841	**29.70/0.843**
CRF25	27.15/0.759	27.23/0.761	27.11/0.760	27.22/0.760	27.24/0.762	**27.28/0.763**
CRF35	24.03/0.644	24.12/0.646	24.05/0.641	24.11/0.644	24.12/0.645	**24.22/0.646**

Conclusion

- Similar to Transformer in high-level tasks, it also shows its great power in low-level tasks. However, it may need some adaptation. Directly applying Transformer to low-level vision may cause performance drops.
  - Hybrid network design (CNN + Transformer)
  - Cross-scale feature learning and fusion
  - Carefully designed attention for specific task
- Transformer/Hybrid based models maybe the future of network design for low-level tasks, but currently it has some deployment issues.

Future Works

- Designing more efficient and hardware friendly model for low-level tasks, especially for Transformer/Hybrid-based models.
- Exploring network design for content creation (high-level + low-level), especially for multi-modal diffusion-based generation.

Thanks

Q&A

E-mail: huayan@microsoft.com