https://doi.org/10.2352/EI.2023.35.7.IMAGE-272

Light-Weight Recurrent Network for Real-Time Video Super-Resolution [*]

Tianqi Wang, Jan P. Allebach; Purdue University, West Lafayette, IN.
Qian Lin; HP Inc., Palo Alto, CA.

Abstract

Real-time video super-resolution (VSR) has been considered a promising solution to improving video quality for video conferencing and media video playing, which requires low latency and short inference time. Although state-of-the-art VSR methods have been proposed with well-designed architectures, many of them are not feasible to be transformed into a real-time VSR model because of vast computation complexity and memory occupation. In this work, we propose a light-weight recurrent network for this task, where motion compensation offset is estimated by an optical flow estimation network, features extracted from the previous high-resolution output are aligned to the current target frame, and a hidden space is utilized to propagate long-term information. We show that the proposed method is efficient in real-time video super-resolution. We also carefully study the effectiveness of the existence of an optical flow estimation module in a light-weight recurrent VSR model and compare two ways of training the models. We further compare four different motion estimation networks that have been used in light-weight VSR approaches and demonstrate the importance of reducing information loss in motion estimation.

Introduction

Video super-resolution (VSR) is the process of reconstructing high-resolution frames from a sequence of low-resolution frames. There has been a trend of increasing usage of video conferencing and video telephony for remote communication in both professional and private life. In these scenarios, as well as media video playing, low-resolution videos exist, which hinder effective communication and degrade the video watching experience because of small and blurry target regions. Real-time VSR is a promising solution to this problem. However, this task is challenging due to the trade-off between model capacity and network latency.

In recent years, state-of-the-art VSR approaches are adopting CNN [1, 2, 3] and Transformer [4, 5, 6] architectures. Although having different backbone structures, CNN-based models and Transformer-based models are both exploiting correspondence between video frames. Because CNN-based models have inherently inductive biases of locality and each convolution step can only focus on the area near the central position [4, 6], the alignment module is essential for performance improvement [3, 1]. While in a Transformer attention window, there is no locality inductive bias, and therefore Transformer can handle misalignment within the attention window [6]. When the pixel movement is large, Transformer-based models also need alignment modules [4, 6].

Many state-of-the-art models, both CNN-based and Transformer-based, are highly computationally expensive and are thus not feasible for real-time VSR applications. Another branch of VSR explores efficient VSR modules, including recurrent latent space [7, 8], light-weight alignment network [9], and deformable attention [10]. These works achieve real-time video reconstruction on a high-end GPU. In order to achieve real-time VSR on lower-end and mobile devices, experiments have been conducted on network pruning and neural architecture search [11, 12, 13, 14].

In this work, we propose a light-weight recurrent model for real-time video super-resolution, investigate the effectiveness of the existence of the motion estimation module in a lightweight VSR network, and compare different motion estimation networks and two strategies to train a network with an alignment module. Specifically, the proposed VSR model has two main networks, the flow estimation network, Fnet, and the super-resolution network, SRnet. The model warps features extracted from the previous high-resolution output by a motion compensation offset estimated by Fnet. The transformed high-resolution features, extracted low-resolution features from the current target frame, and the hidden state are then concatenated and fed to the SRnet to reconstruct the current target frame.

We make the following contributions: 1) We propose an efficient recurrent video super-resolution network. It aligns features for better temporal correspondence and is trained by a two-term loss function; 2) We demonstrate that the SpyNet flow estimation network in a VSR task performs better if guided by a loss term measuring the difference between the warped neighbor frame and the target frame; 3) Different flow estimation networks have been used in VSR methods, of which each has its advantages. We compare SpyNet [15], U-net, modified U-net, and CNN, and show that a skip concatenation improves the performance and can deal with the information loss caused by the max-pooling layer in a U-net flow estimation network.

Related Works

VSR deep learning approaches have been adopting sliding-window and recurrent frameworks to input neighboring frames into the models for temporal redundancy exploitation. The sliding window methods [3, 16] employ the LR images within a window for the restoration of the target frame. Sajjadi et al. [17] proposed a recurrent framework, FRVSR, using the previous high-resolution output to ease the burden of frame reconstruction. Besides the previous output, RLSP [8] and RRN [7] also take a latent space as one of the inputs to propagate temporal information in an implicit manner. We choose a recurrent structure to design our

[*] Research supported by HP Inc., Palo Alto, CA 94304

model because of its potential for long-term information propagation and the feasibility of utilizing previous reconstruction results.

To acquire information from frames at a longer temporal distance, some VSR approaches [1, 2, 4] take all frames as input and allow information propagation in both forward and backward directions. Different from the methods that output one frame at a time, these approaches usually output all reconstructed frames together. In this work, we take the current and the past information as input and only allow forward propagation, as low latency is needed in real-time VSR applications.

Finding correspondence and exploiting temporal information has been shown to be essentially important in achieving performance gain for video super-resolution. Several methods [18, 17, 9] use optical flow to estimate motions and perform explicit motion compensation by warping the images using the optical flow. TDAN [19] and EDVR [3] utilize deformable convolution [20] to align images without warping. BasicVSR++ [2] employs optical flow to guide deformable alignment to overcome the problem of training instability introduced by the deformable alignment module. Transformer-based models [4, 5, 6] can implicitly explore spatial and temporal correspondence within an attention window. DAP [10] replaces the attention window with sampled locations to reduce computation complexity and utilize the Transformer attention mechanism to directly retrieve information from neighboring frames without explicit alignment. We adopt motion estimation and explicit alignment in our model.

Proposed Methods

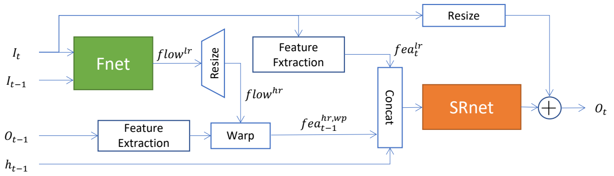


Figure 1. *Architecture of our proposed model. The flow estimation network, Fnet, is a light-weight version of SpyNet [15]. The super-resolution network, SRnet, consists of residual blocks.*

Aiming at discovering frameworks for generic real-time VSR, we confine our architecture design to commonly-adopted light-weight elements. An overview of the proposed network is depicted in Figure 1.

Propagation Propagation specifies what temporal information is accessible for the current reconstruction. The sliding-window methods allow local information propagation within a window. Recurrent frameworks either take local information when the previous outputs are used as input or employ unidirectional propagation implicitly from the past to the current time when the hidden space is also used as input. VSR approaches that take all frames as input can employ bidirectional propagation. The implementation of the bidirectional propagation methods occupies a lot of memory and causes time delays. These methods are more suitable for offline video restoration rather than real-time VSR applications. We take the recurrent structure as our super-resolution network and input the current frame, previous output, and hidden space into the model to exploit current and past information.

Alignment To aggregate information from corresponding locations, recent CNN-based networks usually perform an implicit or explicit alignment. However, many proposed alignment methods are computationally expensive. We experimented with flow-guided deformable alignment [2] and pyramid, cascading, and deformable (PCD) alignment [3] in a light-weight recurrent model. Because inference time is sensitive for real-time VSR, we add the alignment modules and keep the inference time unchanged (37 ms per frame on an Nvidia GeForce RTX 2080) by reducing the number of channels. We observed a 0.48 dB drop for flow-guided deformable alignment and a 0.34 dB drop for PCD alignment. We concluded that when the number of channels is small, computationally expensive alignment cannot offset the negative influence of fewer channels, especially when the number of channels is close to or less than that needed for reconstruction, e.g. 48 for RGB videos when pixel shuffling is used for 4× upsampling VSR.

We perform explicit alignment and use a flow estimation network, Fnet, to compute the optical flow from the previous low-resolution frame to the current low-resolution frame. We choose SpyNet [15] as Fnet and modify it to a light-weight version. We then upsample the optical flow and use it to warp features extracted from previous output O_{t-1}. Note that we estimate the optical flow from the images but use them to warp features. In Chan's experiments [1], image alignment results in a 0.17 dB drop compared to feature alignment. The potential reason is the inaccuracy of optical flow estimation and error propagation. The process to obtain warped high-resolution features from $t-1$ can be formulated as:

$$fea_{t-1}^{hr} = Conv(O_{t-1}) \tag{1}$$

$$flow^{lr} = Fnet(I_{t-1},\ I_t) \tag{2}$$

$$fea_{t-1}^{hr,wp} = Wp(Up(flow^{lr}),\ fea_{t-1}^{hr}) \tag{3}$$

We experiment with three other flow estimation network structures and compare two methods of training a super-resolution network with flow estimation. Details are in the next section.

Reconstruction Residual mapping between layers with identity skip connections preserves the texture information and keeps fluent information flows over long periods. The structure is used in many VSR approaches that output one frame at a time, regardless of variant propagation and alignment methods [7, 9, 3]. We adopt residual blocks in our network as the reconstruction module.

Experiments

Training Datasets and Details

Datasets and Settings We use Vimeo-90K [21] for training, which has a training set of 64,612 7-frame sequences, with fixed resolution 448 × 256. To produce LR images, we blur the HR images with a Gaussian kernel and then downscale the images 4× with nearest-neighbor interpolation. Furthermore, we apply vertical and horizontal flipping as augmentation. We test our model on Vid4 [22], UDM10 [23], and SPMCS [24] datasets.

Training Details The modified SpyNet has 4 resolution scales, $\{H \times W, H/2 \times W/2, H/4 \times W/4, H/8 \times W/8\}$, where we assume that the pixel movement is relatively small for real-time VSR applications and the receptive fields are not necessarily large. There are 3 convolutional layers for each resolution scale. We use 5 residual blocks for the reconstruction module to extract high-frequency features. The channel size in each residual block is set to 64. The mini-batch size is set to 16. We use two loss terms to train our model. The first term is the Charbonnier loss [25] between the ground truth, GT, and the output, O_t, defined by

$$L_{sr} = \sqrt{||GT - O_t||_2^2 + \varepsilon^2} \tag{4}$$

ε is set to 1×10^{-3}. The second term guides the Fnet to estimate optical flow. We calculate the Charbonnier penalty of the difference between the warped previous input, I_{t-1}, and the current input, I_t:

$$L_{flow} = \sqrt{||Wp(I_{t-1}) - I_t||_2^2 + \varepsilon^2} \tag{5}$$

The total training loss is $L = L_{sr} + L_{flow}$. The reasons for using the flow loss term are twofold. One is that the ground truth optical flow is unavailable for this dataset. The other is that the optical flow should be task-specific [21]. Even though an optical flow can be computed by a pre-trained network like the original SpyNet, the optical flow obtained this way is not optimal for the SR task and tends to generate blurry scenes [26]. The optical flow estimation network introduced in an SR task should be re-trained or fine-tuned implicitly or guided by a loss term.

We train our model with the Adam optimizer and set the exponential decay rate for the first and second moment estimates to be $\beta_1 = 0.9$ and $\beta_2 = 0.999$, respectively. The learning rate is initialized as 1×10^{-4} and later down-scaled by a factor of 0.1 after 60 epochs.

Results

We compare our 10-layer model (10 residual blocks in SRnet) with three state-of-the-art VSR approaches: TOFlow [21], FRVSR [17], and RRN [7]. TOFlow adopts SypNet [15] for motion estimation, but it upsamples frames before model inference which is highly inefficient. FRVSR warps images rather than features with motion offsets estimated by a U-net flow estimation network. RRN [7], which does not explicitly align images or features, propagates historical information by leveraging the hidden state. As shown in Table 1, our method outperforms TOFlow by 0.28 dB on the Vid4 dataset and 0.84 dB on the UDM10 dataset with fewer parameters and can run 18 times faster. Compared with FRVSR, our model achieves competitive results while having much lower computational complexity and less runtime. RRN outperforms our approach, but the runtime is about twice as long.

Ablation Study

Alignment and loss function We compare different configurations to validate the effectiveness of the alignment method and the loss term for the optical flow. We train a super-resolution network without flow estimation and alignment, referred to as Method 1, SRnet, which has 67 channels to keep the computational complexity comparable to our proposed model. Method 2 has the same architecture as our proposed network but was trained with a super-resolution loss L_{sr} only. Each of the three models has 5 residual blocks in SRnet. Quantitative and qualitative results are shown in Table 2 and Figure 2, respectively.

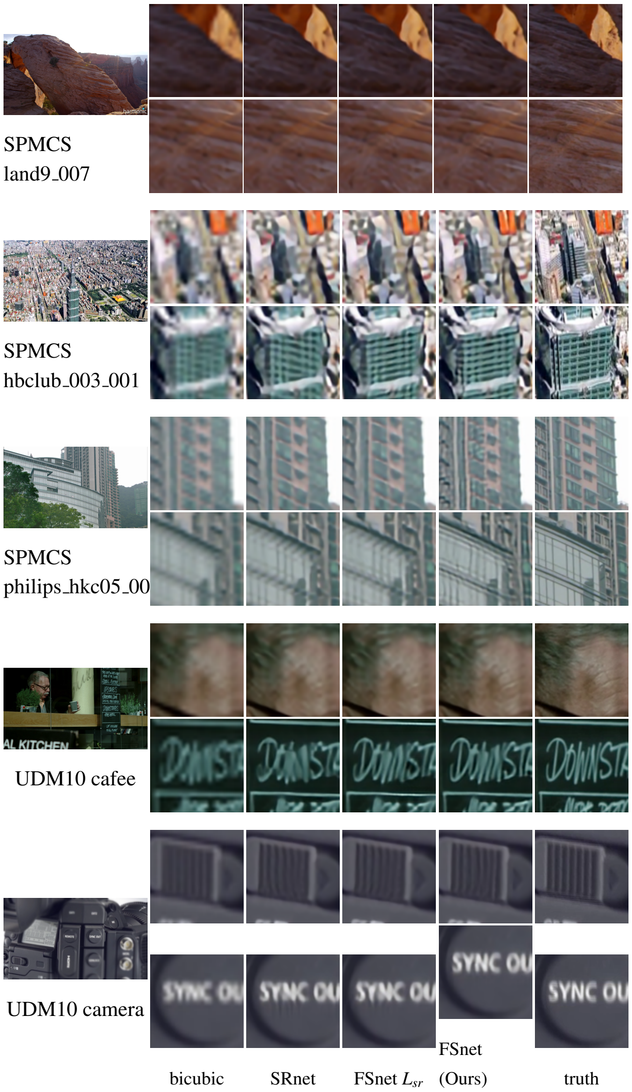


Figure 2. Qualitative results of SRnet, FSnet without flow loss, and our proposed method, FSnet trained with two-term loss, on the UDM10 [23] and SPMCS [24] datasets for 4× VSR.

Our proposed network and training configuration outperforms SRnet by 0.85 dB and FSnet without flow loss L_{flow} by 0.81 dB on the UDM10 dataset. Visually, the proposed method generates sharper and more detailed results.

Flow estimation network In this work, we estimate the optical flow by SpyNet [15], which is also used in TOFlow [21]. SpyNet first learns to generate an optical flow at the lowest resolution (H/8 × W/8 in this work), upsamples this optical flow, and learns a residual to modify the upsampled flow. It then repeats this process until generating an optical flow at the full resolution (H × W). It can be viewed as coarse-to-fine optical flow learning. Both in the SpyNet and in our network, there are operations that warp the previous frame using the upsampled optical flow, where the upsampling is performed by bilinear interpolation. This could potentially cause detail loss because of the inaccuracy of flow introduced by the upsampling method. FRVSR [17] and EGVSR [9] adopt the U-net structure for flow estimation. Similar

Table 1. Quantitative comparison (PSNR and SSIM) on Vid4 and UDM10 for $4\times$ VSR. Y and RGB indicate the evaluation on the luminance channel or RGB channels, respectively. Runtime and MACs (multiply-accumulate computations) are evaluated for an LR image of size 180×320 on an Nvidia GeForce RTX 2080.

Method	Bicubic	TOFlow [21]	FRVSR[17]	RRN [7]	Ours (10-64)
# Param. [M]	N/A	1.4	5.1	3.4	1.1
MACs [G]	N/A	135.9	352.1	193.9	56.0
Runtime [ms]	N/A	1070	126	100	55
Vid4 (Y)	21.80/0.5426	25.85/0.7659	26.48/0.8104	27.41/0.8466	26.13/0.7826
Vid4 (RGB)	20.37/0.5106	24.39/0.7438	25.01/0.7917	25.91/0.8288	24.66/0.7604
UDM10 (Y)	28.47/0.8523	36.26/0.9438	37.09/0.9522	38.74/0.9642	37.11/0.9506
UDM10 (RGB)	27.05/0.8267	34.46/0.9298	35.39/0.9403	36.83/0.9530	35.24/0.9363

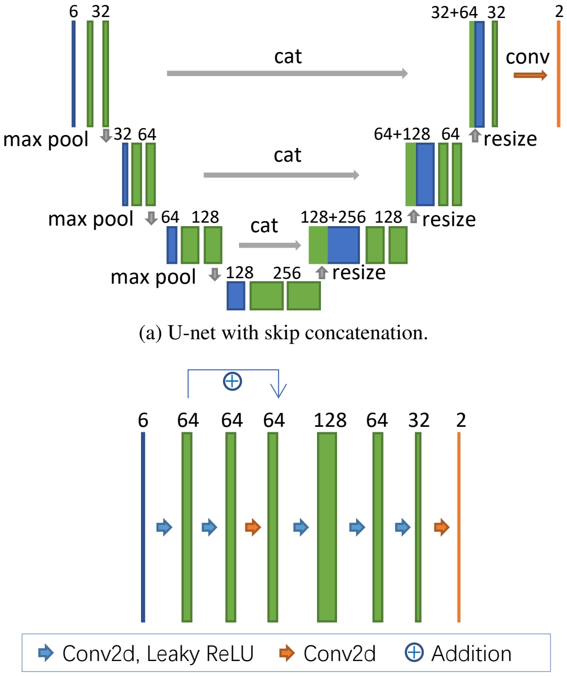
Table 2. Quantitative comparison (PSNR(dB) and SSIM) of different configurations of VSR network for $4\times$ VSR. Results are tested on RGB channels. Red text indicates the best performance.

Method	1. SRnet	2. FSnet	3. FSnet (Ours)
# channel	67	64	64
loss	L_{sr}	L_{sr}	$L_{sr} + L_{flow}$
# Param. [M]	0.60	0.70	0.70
MACs [G]	34.4	34.7	34.7
Runtime [ms]	49	49	49
Vid4	23.72/0.7042	23.77/0.7083	23.92/0.7354
SPMCS	27.41/0.7981	27.35/0.7974	27.49/0.8054
UDM10	33.75/0.9221	33.79/0.9211	34.60/0.9301

to SpyNet, the U-net structure has a large receptive field at the low-resolution convolutional layers. Another advantage of U-net is that more neurons and channels can be used while maintaining comparable inference time or computation complexity. However, because of the existence of the max-pooling layer, details may be lost in the optical flow estimation. A concatenation operation has the potential to deal with this information loss. Another optical flow estimation network contains only convolutional layers at full resolution.

Table 3. Quantitative comparison (PSNR(dB) and SSIM) of four optical flow estimation networks for $4\times$ VSR. Results are tested on RGB channels. Red text indicates the best performance.

Fnet	U-net	U-net_cat	Conv-net	SpyNet [15]
# Param. [M]	2.29	2.49	0.79	0.70
FLOPs [G]	39.1	40.71	45.67	34.7
Runtime [ms]	49	50	50	49
Vid4	23.70/0.7032	24.16/0.7386	23.89/0.7147	23.92/0.7354
SPMCS	27.41/0.7975	27.46/0.8042	27.44/0.7982	27.49/0.8054
UDM10	33.74/0.9219	34.59/0.9306	33.87/0.9229	34.60/0.9301



(a) U-net with skip concatenation.

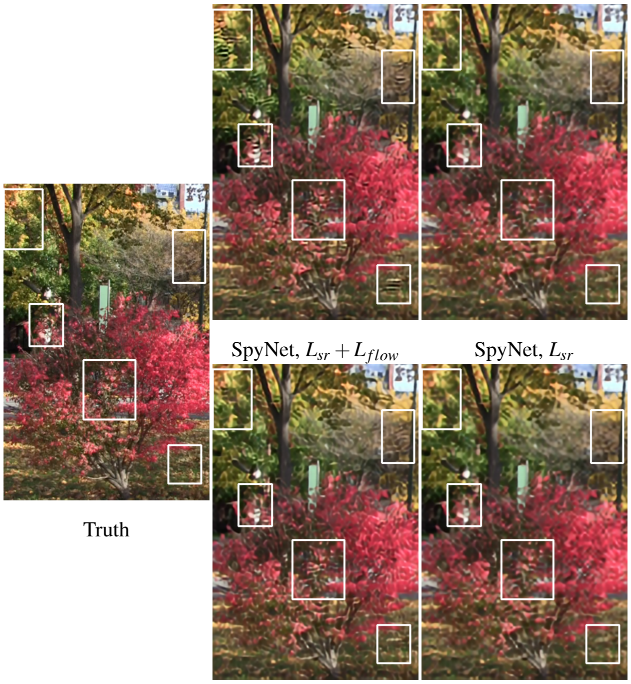
(b) Conv-net

Figure 3. Architectures of two optical flow estimation networks: (a) U-net with skip concatenation and (b) Conv-net.

We compare 4 types of flow estimation networks in the VSR model: SpyNet, U-net, U-net with skip concatenation, and Conv-net at the full resolution. Figure 3 shows the architectures of the last two networks. The four networks' configurations are chosen to have similar inference times using the PyTorch framework tested on an Nvidia GeForce RTX 2080. The feature extraction module and reconstruction module are the same for the four flow estimation networks, where the SRnet has 5 residual blocks. They are all trained with a two-term loss $L = L_{sr} + L_{flow}$. As shown in Table 3, U-net with skip concatenation outperforms U-net by 0.46 dB on Vid4 and 0.85 dB on UDM10, with a slight increase in the computational cost. This comparison result backs our hypothesis that a concatenation operation can overcome the disadvantage introduced by the max pooling layer in the U-net. SpyNet produces the best results on SPMCS, the best PSNR on UDM10, and nearly the best SSIM on UDM10, and the second second-best results on Vid4, while having the least number of parameters among the four networks.

Failure Cases

Compared with other Fnet and loss function choices in our experiment, VSR models having SpyNet and U-net with skip concatenation as Fnet trained with two-term loss functions achieve higher PSNR and SSIM. However, these two configurations produce VSR results with stripe artifacts on a testing video ”foliage”



SpyNet, $L_{sr} + L_{flow}$ SpyNet, L_{sr}

Truth

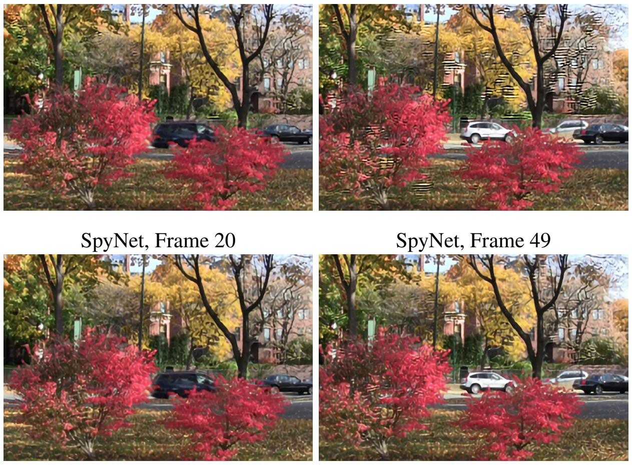
U-net_cat, $L_{sr} + L_{flow}$ U-net, $L_{sr} + L_{flow}$

Figure 4. *Visual results of the testing video "foliage" (frame 30) in the Vid4 dataset. VSR models with SpyNet and U-net_cat as Fnet generate stripe artifacts.*

in the Vid4 dataset (Figure 4). The super-resolution "foliage" video produced by our proposed model, which has SpyNet as Fnet, has stronger stripe artifacts. We also observed that this artifact was mild in the first few frames, then it become stronger and stronger in time (Figure 5). Potential causes of these artifacts are the inaccuracy of motion estimation for videos having drastic luminance changes and the error propagation and accumulation in time in a recurrent neural network. Note that when SpyNet is trained only with a super-resolution loss (Equation 4), the VSR model does not generate unwanted artifacts. U-net without skip concatenation also does not generate this type of artifact.

Conclusion

In this paper, we proposed a recurrent network for real-time video super-resolution. It extracts features from the previous high-resolution output and warps them with motion compensation offsets computed by a light-weight version of SpyNet. Then the SR network takes low-resolution features, warped high-resolution features, and the latent space to reconstruct the current frame. Experiments on various benchmark datasets show that our model is efficient for video super-resolution tasks. The ablation studies show the effectiveness of our model choice and training strategy. However, one failure case limits the application scenarios of our proposed method. Our experiments with four different flow estimation networks show the importance of reducing information loss in the network and that a concatenation operation can overcome the disadvantage introduced by the max pooling layer in the U-net.



SpyNet, Frame 20 SpyNet, Frame 49

U-net_cat, Frame 20 U-net_cat, Frame 49

Figure 5. *Super-resolution results for Frame 20 and Frame 49 of the "foliage" video in the Vid4 dataset generated by VSR models with SpyNet and U-net_cat as Fnet. The stripe artifacts get stronger from Frame 20 to Frame 49.*

References

[1] 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," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4947–4956, 2021.

[2] K. C. Chan, S. Zhou, X. Xu, and C. C. Loy, "BasicVSR++: Improving video super-resolution with enhanced propagation and alignment," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5972–5981, 2022.

[3] X. Wang, K. C. Chan, K. Yu, C. Dong, and C. Change Loy, "EDVR: Video restoration with enhanced deformable convolutional networks," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1954–1963, 2019.

[4] J. Liang, J. Cao, Y. Fan, K. Zhang, R. Ranjan, Y. Li, R. Timofte, and L. Van Gool, "VRT: A video restoration transformer," *arXiv preprint arXiv:2201.12288*, 2022.

[5] J. Liang, Y. Fan, X. Xiang, R. Ranjan, E. Ilg, S. Green, J. Cao, K. Zhang, R. Timofte, and L. V. Gool, "Recurrent video restoration transformer with guided deformable attention," *Advances in Neural Information Processing Systems*, vol. 35, 2022.

[6] S. Shi, J. Gu, L. Xie, X. Wang, Y. Yang, and C. Dong, "Rethinking alignment in video super-resolution transformers," *Advances in Neural Information Processing Systems*, vol. 35, pp. 36081–36093, 2022.

[7] T. Isobe, F. Zhu, X. Jia, and S. Wang, "Revisiting temporal modeling for video super-resolution," *Proceedings of the British Machine Vision Conference*, 2020.

[8] D. Fuoli, S. Gu, and R. Timofte, "Efficient video super-resolution through recurrent latent space propagation," *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshop*, pp. 3476–3485, 2019.

[9] Y. Cao, C. Wang, C. Song, Y. Tang, and H. Li, "Real-time super-resolution system of 4k-video based on deep learning," *Proceedings of the IEEE International Conference on Application-specific Systems, Architectures and Processors*, pp. 69–76, 2021.

[10] D. Fuoli, M. Danelljan, R. Timofte, and L. Van Gool, "Fast online

video super-resolution with deformable attention pyramid," *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*.

[11] H. Kim, S. Hong, B. Han, H. Myeong, and K. M. Lee, "Fine-grained neural architecture search," *arXiv preprint arXiv:1911.07478*, 2019.

[12] S. Liu, C. Zheng, K. Lu, S. Gao, N. Wang, B. Wang, D. Zhang, X. Zhang, and T. Xu, "EVSRNet: Efficient video super-resolution with neural architecture search," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*.

[13] Z. Zhan, Y. Gong, P. Zhao, G. Yuan, W. Niu, Y. Wu, T. Zhang, M. Jayaweera, D. Kaeli, B. Ren, X. Lin, and Y. Wang, "Achieving on-mobile real-time super-resolution with neural architecture and pruning search," *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4801–4811, 2021.

[14] Y. Wu, Y. Gong, P. Zhao, Y. Li, Z. Zhan, W. Niu, H. Tang, M. Qin, B. Ren, and Y. Wang, "Compiler-aware neural architecture search for on-mobile real-time super-resolution," *Proceedings of the European Conference on Computer Vision*, pp. 92–111, 2022.

[15] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2720–2729, 2017.

[16] T. Isobe, S. Li, X. Jia, S. Yuan, G. Slabaugh, C. Xu, Y.-L. Li, S. Wang, and Q. Tian, "Video super-resolution with temporal group attention," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8005–8014, 2020.

[17] M. S. Sajjadi, R. Vemulapalli, and M. Brown, "Frame-recurrent video super-resolution," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6626–6634, 2018.

[18] J. Caballero, C. Ledig, A. Aitken, A. Acosta, J. Totz, Z. Wang, and W. Shi, "Real-time video super-resolution with spatio-temporal networks and motion compensation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4778–4787, 2017.

[19] Y. Tian, Y. Zhang, Y. Fu, and C. Xu, "TDAN: Temporally-deformable alignment network for video super-resolution," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3357–3366, 2020.

[20] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 764–773, 2017.

[21] 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.

[22] C. Liu and D. Sun, "A Bayesian approach to adaptive video super resolution," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 209–216, 2011.

[23] P. Yi, Z. Wang, K. Jiang, J. Jiang, and J. Ma, "Progressive fusion video super-resolution network via exploiting non-local spatio-temporal correlations," *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3106–3115, 2019.

[24] X. Tao, H. Gao, R. Liao, J. Wang, and J. Jia, "Detail-revealing deep video super-resolution," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4482–4490, 2017.

[25] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud, "Two deterministic half-quadratic regularization algorithms for computed imaging," *Proceedings of IEEE International Conference on Image Processing*, vol. 2, pp. 168–172, 1994.

[26] K. C. Chan, X. Wang, K. Yu, C. Dong, and C. C. Loy, "Understanding deformable alignment in video super-resolution," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 2, pp. 973–981, 2021.

Author Biography

Tianqi Wang is a Ph.D. candidate working with Professor Jan Allebach in Electrical and Computer Engineering at Purdue University. Tianqi graduated with her B.S. in Biomedical Engineering from Northeastern University (China), and her M.S. in Civil Engineering from Purdue University. She is broadly interested in computer vision, machine learning, and data mining. Her Ph.D. research has an array of applications including customer data mining, video super-resolution, and document classification.