Towards High-Quality and Efficient Video Super-Resolution via Spatial-Temporal Data Overfitting

Gen Li[1,†], Jie Ji[1,†], Minghai Qin[†], Wei Niu[2], Bin Ren[2], Fatemeh Afghah[1], Linke Guo[1], Xiaolong Ma[1]
[1]Clemson University [2]William & Mary
{gen,xiaolom}@clemson.edu

Abstract

As deep convolutional neural networks (DNNs) are widely used in various fields of computer vision, leveraging the overfitting ability of the DNN to achieve video resolution upscaling has become a new trend in the modern video delivery system. By dividing videos into chunks and overfitting each chunk with a super-resolution model, the server encodes videos before transmitting them to the clients, thus achieving better video quality and transmission efficiency. However, a large number of chunks are expected to ensure good overfitting quality, which substantially increases the storage and consumes more bandwidth resources for data transmission. On the other hand, decreasing the number of chunks through training optimization techniques usually requires high model capacity, which significantly slows down execution speed. To reconcile such, we propose a novel method for high-quality and efficient video resolution upscaling tasks, which leverages the spatial-temporal information to accurately divide video into chunks, thus keeping the number of chunks as well as the model size to minimum. Additionally, we advance our method into a single overfitting model by a data-aware joint training technique, which further reduces the storage requirement with negligible quality drop. We deploy our models on an off-the-shelf mobile phone, and experimental results show that our method achieves real-time video super-resolution with high video quality. Compared with the state-of-the-art, our method achieves 28 fps streaming speed with 41.6 PSNR, which is 14× faster and 2.29 dB better in the live video resolution upscaling tasks. Code available in https://github.com/coulsonlee/STDO-CVPR2023.git.

1. Introduction

Being praised by its high image quality performance and wide application scenarios, deep learning-based super-resolution (SR) becomes the core enabler of many incredible, cutting-edge applications in the field of image/video reparation [10, 11, 39, 40], surveillance system enhancement [9], medical image processing [35], and high-quality video live streaming [20]. Distinct from the traditional methods that adopt classic interpolation algorithms [15,45] to improve the image/video quality, the deep learning-based approaches [10, 11, 21, 24, 28, 40, 44, 47, 57, 60] exploit the advantages of learning a mapping function from low-resolution (LR) to high-resolution (HR) using external data, thus achieving better performance due to better generalization ability when meeting new data.

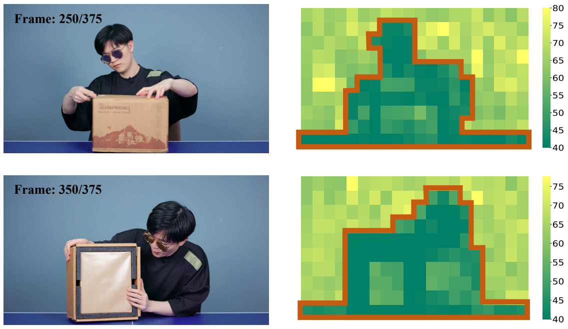


Figure 1. Patch PSNR heatmap of two frames in a 15s video when super-resolved by a general WDSR model. A clear boundary shows that PSNR is strongly related to video content.

Such benefits have driven numerous interests in designing new methods [5, 17, 50] to deliver high-quality video stream to users in the real-time fashion, especially in the context of massive online video and live streaming available. Among this huge family, an emerging representative [13,16,31,38] studies the prospect of utilizing *SR model* to upscale the resolution of the LR video in lieu of transmitting the HR video directly, which in many cases, consumes tremendous bandwidth between servers and clients [19]. One practical method is to deploy a pretrained SR model on the devices of the end users [25, 54], and perform resolution upscaling for the transmitted LR videos, thus obtaining HR videos without causing bandwidth congestion. However, the deployed SR model that is trained with limited data usually suffers from limited generalization abil-

† Equal Contribution.

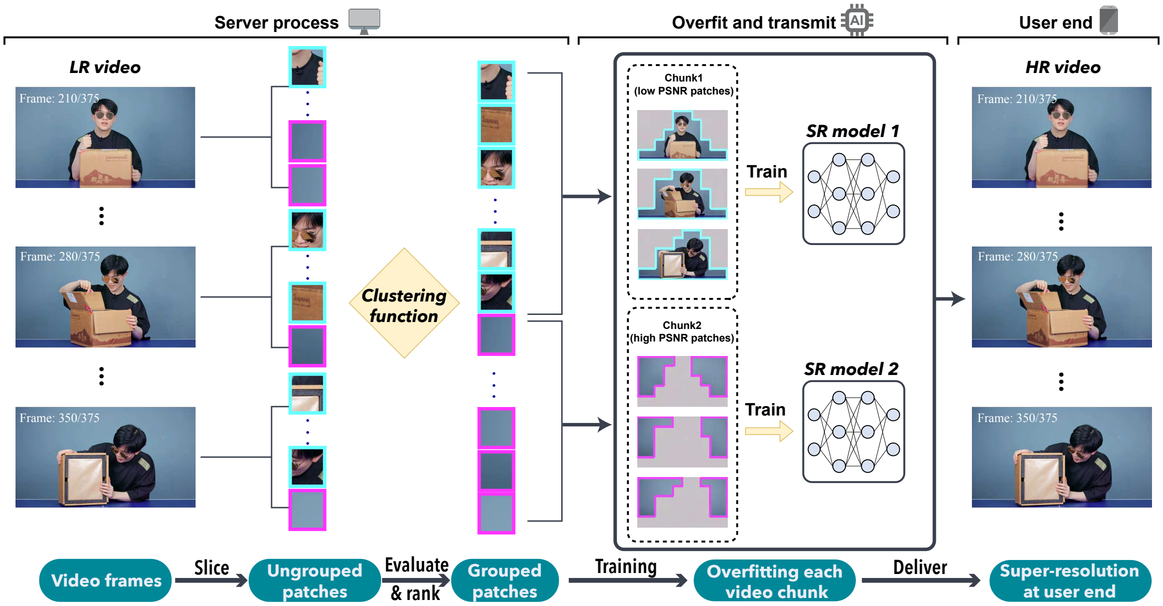


Figure 2. Overview of the proposed STDO method. Each video frame is sliced into patches, and all patches across time dimension are divided and grouped into chunks. Here we set the number of chunks to 2 for clear illustration. Then each chunk is overfitted by independent SR models, and delivered to end-user for video super-resolution.

ity, and may not achieve good performance at the presence of new data distribution [55]. To overcome this limitation, new approaches [4, 8, 20, 30, 51, 53, 55] exploit the overfitting property of DNN by training an SR model for each video chunk (i.e., a fragment of the video), and delivering the video alongside the corresponding SR models to the clients. This trade-off between model expressive power and the storage efficiency significantly improves the quality of the resolution upscaled videos. However, to obtain better overfitting quality, more video segments are expected, which notably increase the data volume as well as system overhead when processing the LR videos [55]. While advanced training techniques are proposed to reduce the number of SR models [30], it still requires overparameterized SR backbones (e.g., EDSR [28]) and handcrafted modules to ensure sufficient model capacity for the learning tasks, which degrades the execution speed at user-end when the device is resource-constraint.

In this work, we present a novel approach towards high-quality and efficient video resolution upscaling via **S**patial-**T**emporal **D**ata **O**verfitting, namely **STDO**, which for the first time, utilizes the spatial-temporal information to accurately divide video into chunks. Inspired by the work proposed in [1, 14, 23, 46, 58] that images may have different levels of intra- and inter-image (i.e., within one image or between different images) information density due to varied texture complexity, we argue that the unbalanced information density within or between frames of the video universally exists, and should be properly managed for data overfitting. Our preliminary experiment in Figure 1 shows that the PSNR values at different locations in a video frame forms certain pattern regarding the video content, and exhibits different patterns along the timeline. Specifically, at the server end, each frame of the video is evenly divided into patches, and then we split all the patches into multiple chunks by PSNR regarding all frames. Independent SR models will be used to overfit the video chunks, and then delivered to the clients. Figure 2 demonstrates the overview of our proposed method. By using spatial-temporal information for data overfitting, we reduce the number of chunks as well as the overfitting models since they are bounded by the nature of the content, which means our method can keep a minimum number of chunks regardless the duration of videos. In addition, since each chunk has similar data patches, we can actually use smaller SR model without handcrafted modules for the overfitting task, which reduces the computation burden for devices of the end-user. Our experimental results demonstrate that our method achieves real-time video resolution upscaling from 270p to 1080p on an off-the-shelf mobile phone with high PSNR.

Note that STDO encodes different video chunks with independent SR models, we further improve it by a **J**oint training technique (**JSTDO**) that results in one single SR model for all chunks, which further reduces the storage requirement. We design a novel data-aware joint training technique, which trains a single SR model with more data from higher information density chunks and less data from their counterparts. The underlying rationale is consistent with the discovery in [46, 58], that more informative data contributes majorly to the model training. We summarize our contributions as follows:

- We discover the unbalanced information density within video frames, and it universally exists and constantly changes along the video timeline.
- By leveraging the unbalanced information density in the video, we propose a spatial-temporal data overfitting method STDO for video resolution upscaling, which achieves outperforming video quality as well as real-time execution speed.
- We propose an advanced data-aware joint training technique which takes different chunk information density into consideration, and reduces the number of SR models to a single model with negligible quality degradation.
- We deploy our models on an off-the-shelf mobile phone, and achieve real-time super-resolution performance.

2. Related Works

2.1. Single Image Super Resolution (SISR)

For SISR tasks, SRCNN [10] is the pioneer of applying DNN to image super resolution. Then, followed by FSRCNN [11] and ESPCN [40], both of them make progress in efficiency and performance. After this, with the development of deep neural networks, more and more network backbones are used for SISR tasks. For example, VDSR [21] uses the VGG [41] network as the backbone and adds residual learning to further improve the effectiveness. Similarly, SRResNet [24] proposed a SR network using ResNet [18] as a backbone. EDSR [28] removes the batch norm in residual blocks by finding that the use of batch norm will ignore absolute differences between image pixels (or features). WDSR [57] finds that ReLU will impede information transfer, so the growth of the the number of filters before ReLU increases the width of the feature map. With the emergence of channel attention mechanisms networks represented by SENet [36], various applications of attention mechanisms poured into the area of image super resolution [7, 34, 60, 61]. After witnessing the excellent performance of transformer [12] in the field of computer vision, more and more researchers apply various vision transformer models into image super resolution tasks [3, 27, 32].

2.2. Video Super Resolution (VSR)

The VSR methods mainly learn from SISR framework [29]. Some aforementioned works like EDSR and WDSR are all present results on VSR. Some of the current VSR works perform alignment to estimate the motions between images by computing optical flow by DNNs [2, 22, 39, 42]. The Deformable convolution (DConv) [6] was first used to deal with geometric transformation in vision tasks because the sampling operation of CNN is fixed. TDAN [43] applies DConv to align the input frames at the feature level, which avoids the two-stage process used in previous optical flow based methods. EDVR [49] uses their proposed Pyramid, Cascading and Deformable convolutions (PCD) alignment module and the temporal-spatial attention (TSA) fusion module to further improve the robustness of alignment and take account of the visual informativeness of each frame. Other works like DNLN [48] and D3Dnet [56] also apply Dconv in their model to achieve a better alignment performance.

2.3. Content-Aware DNN

It is impractical to develop a DNN model to work well on all the video from Internet. NAS [55] was the first proposed video delivery framework to consider using DNN models to overfit each video chunk to guarantee reliability and performance. Other livestreaming and video streaming works [4, 8, 20, 51, 53] leverage overfitting property to ensure excellent performance at the client end. [20] proposes a live video ingest framework, which adds an online learning module to the original NAS [55] framework to further ensure quality. NEMO [53] selects key frames to apply super-resolution. This greatly reduces the amount of computation on the client sides. CaFM [30] splits a long video into different chunks by time and design a handcrafted layer along with a joint training technique to reduce the number of SR models and improve performance. EMT [26] proposes to leverage meta-tuning and challenge patches sampling technique to further reduce model size and computation cost. SRVC [19] encodes video into content and time-varying model streams. Our work differentiates from these works by taking spatial information as well as temporal information into account, which exhibits better training effects for the overfitting tasks.

3. Proposed Method

3.1. Motivation

To tackle the limited generalization ability caused by using only one general SR model to super-resolve various videos, previous works [20, 30, 55] split the video by time and train separate SR models to overfit each of the video chunks. With more fine-grained video chunks over time, the better overfitting quality can be obtained, which makes these approaches essentially a trade-off between model expressive power and the storage efficiency. For a specific video, more chunks will surely improve the overfitting quality, but it also inevitably increases the data volume as well as system overhead when performing SR inference.

In the aforementioned methods, images are stacked according to the timeline to form the video. However, patches have spatial location information [37], and these patches are fed into the neural network indiscriminately for training, which may cause redundancy that contradicts with overfitting property. As illustrated in Figure 1, when using a general SR model to super-resolve an LR video, the values of PSNR at different patch locations form a clear boundary, and are strongly related to the content of the current video frame (i.e., spatial information). Meanwhile, diverse boundary patterns can be seen in different frames (i.e., temporal information). This observation motivates us to use the spatial-temporal information to accurately divide video into chunks, which exhibits a different design space to overfit video data. With the different levels of information density within each patches, the key insight is to cluster patches that has similar texture complexity across all frames, and use one SR model to overfit each patch group. In this way, the number of video chunks and their associated SR models are effectively reduced, which improves the encoding efficiency regardless the duration of videos. Meanwhile, a compact SR model can be adopted without causing quality degradation because each SR model only overfits one specific video content with similar texture complexity. Additionally, when the spatial-temporal data is properly scheduled, our method can be extended to a joint training manner which generates a single SR model for the entire video.

3.2. Spatial-Temporal Data Overfitting

In this section, we introduce a novel spatial-temporal data overfitting approach, STDO, which efficiently encodes HR videos into LR videos and their corresponding SR models, and achieves outperforming super-resolution quality & speed at user end.

Suppose the video time length is T. General method to train an SR model would firstly divide the video into frames, and slice each frame into multiple non-overlapping image patches. All patches across all dimensions such as their locations in the frame or time compose a complete video. For a given video with the dimension $W \times H$, and the desired patch size $w \times h$, the patch is denoted as $P_{i,j,t}$, where $i \in [0, I)$, $j \in [0, J)$, and $t \in [0, T)$. Note that $I = \lfloor \frac{W}{w} \rfloor$ and $J = \lfloor \frac{H}{h} \rfloor$ are integer numbers, then the training set for this specific video is $\mathcal{D} = \{P_n\}_0^N$ where $N = I \times J \times T$ is the total number of patches.

Note that $\mathcal{D}$ contains all patches across all dimensions. We use a pretrained SR model $f_0(\cdot)$ to super-resolve all of the LR patches and compute their PSNRs with the HR patches. As illustrated in Figure 1, we find that the distribution of the PSNR is usually not uniform, and shows a clear boundary regarding the content of the video. We divide the training set $\mathcal{D}$ into multiple chunks by grouping patches with similar PSNRs, and form a set of chunks as $\Omega = \{\hat{\mathcal{D}}_0, \hat{\mathcal{D}}_1, \ldots, \hat{\mathcal{D}}_k\}$, in which

$$\hat{\mathcal{D}}_k = \{P_n | P_n \in \mathcal{D}, PSNR(f_0(P_n)) \in [\lambda_{k1}, \lambda_{k2})\}, \quad (1)$$

where λ is the threshold. We set the first chunk $\hat{\mathcal{D}}_0$ to be group of the patches with lowest PSNRs, and we list all chunks in ascending order, which means $\hat{\mathcal{D}}_k$ to be the patches with the highest PSNRs. In this way, we separate training data by their spatial-temporal information in a one-shot manner, which is usually done in seconds and can be considered negligible compared to the training process. In this paper, we empirically divide $\mathcal{D}$ evenly. Finally, we train an SR model $f_{sr_k}(\mathbf{w}_k; \hat{\mathcal{D}}_k)$ to overfit each video chunk $\hat{\mathcal{D}}_k$. Experimental results indicate better performance on both video quality and execution speed. Our empirical analysis is that by accurately identifying and grouping the data with similar information density (i.e, texture complexity) into chunks, each SR model becomes easier to "memorize" similar data in an overfitting task, and subsequently demands smaller SR models that can be executed in a real-time fashion.

Table 1. Video super-resolution results comparison of vanilla STDO training and using *only one* model from STDO trained with the most informative chunk $\hat{\mathcal{D}}_0$ and least informative chunk $\hat{\mathcal{D}}_k$, respectively. We also include the video super-resolution results with one model trained with all data [54].

Model	Method	vlog-15s			vlog-45s		
		×2	×3	×4	×2	×3	×4
WDSR	awDNN [54]	49.24	45.30	43.33	48.02	44.16	42.19
	STDO	50.58	46.43	44.62	49.76	45.95	43.99
	STDO$\vert_{\hat{\mathcal{D}}_0}$	50.42	45.99	44.18	49.51	45.63	43.75
	STDO$\vert_{\hat{\mathcal{D}}_k}$	46.89	42.63	40.25	44.89	41.07	38.87

3.3. Data-Aware Joint Training

In Section 3.2, our method significantly reduces the number of video chunks and overfitting SR models by effectively utilizing spatial-temporal information of each patch in the video. In this section, we extend our method by generating a single SR model for the entire video, which further reduces the storage requirement with negligible quality drop. From the set of chunks $\Omega \in \mathbb{R}^k$ and all SR models, we demonstrate PSNR in Table 1 by using *only one* SR model to super-resolve the entire video. Somehow surprisingly, we find out that using the model trained with $\hat{\mathcal{D}}_0$ (i.e., the most informative chunk) experiences a moderate quality drop, and achieves similar or higher PSNR compared to the model trained with all patches. Meanwhile, the model trained with $\hat{\mathcal{D}}_k$ has a severe quality degradation. We argue that low PSNR patches usually contain rich features, and contribute more to the model learning, which improves the generalization ability to the less informative data [46, 58]. Motivated by the above observation, we propose a joint training technique, which carefully schedules the spatial-temporal data participated in training to train a single SR model that overfits the entire video. Concretely, we keep all patches for $\hat{\mathcal{D}}_0$, and remove the entire $\hat{\mathcal{D}}_k$. For the rest of the chunks, we randomly sample a portion of the patches from each chunk, while gradually decreasing the proportion

Table 2. Comparison results of STDO with different data overfitting methods on different SR backbones.

	Data		**game-15s**			**inter-15s**			**vlog-15s**	
Model	**Scale**	×2	×3	×4	×2	×3	×4	×2	×3	×4
ESPCN	awDNN [54]	37.94	32.85	29.97	40.43	35.36	29.91	46.41	42.90	39.65
	NAS [55]	37.58	32.71	30.59	40.62	35.42	30.43	46.53	43.01	39.98
	CaFM [30]	38.07	33.14	30.96	40.71	35.54	30.47	47.02	43.20	40.16
	STDO	**38.61**	**33.57**	**31.30**	**42.65**	**35.63**	**30.63**	**47.11**	**43.25**	**40.73**
SRCNN	awDNN [54]	36.08	31.94	29.90	40.46	33.95	28.78	46.69	42.41	39.71
	NAS [55]	36.27	32.08	29.94	40.70	34.01	28.84	46.78	42.53	39.76
	CaFM [30]	36.63	32.21	29.98	40.76	34.08	29.93	46.98	42.62	39.81
	STDO	**37.59**	**32.67**	**30.64**	**42.28**	**34.26**	**30.05**	**47.06**	**42.78**	**39.90**
VDSR	awDNN [54]	41.27	35.03	32.16	44.16	35.99	30.65	48.18	43.03	41.07
	NAS [55]	42.53	35.97	33.86	44.71	36.57	31.05	48.49	43.41	41.33
	CaFM [30]	43.02	36.17	33.98	44.85	36.46	31.08	48.61	43.62	41.49
	STDO	**43.56**	**36.71**	**35.02**	**45.16**	**36.81**	**33.43**	**48.75**	**43.82**	**41.71**
EDSR	awDNN [54]	42.24	35.88	33.44	43.06	37.89	34.94	48.87	44.51	42.58
	NAS [55]	42.82	36.42	34.00	45.06	38.38	35.47	49.10	44.80	42.83
	CaFM [30]	43.13	37.04	34.47	45.35	38.66	35.70	49.30	45.03	43.12
	STDO	**44.93**	**37.80**	**35.47**	**45.91**	**39.26**	**36.76**	**50.24**	**45.68**	**43.46**
WDSR	awDNN [54]	43.36	37.12	34.62	44.83	39.05	35.23	49.24	45.30	43.33
	NAS [55]	44.17	38.23	36.02	45.43	39.71	36.54	49.98	45.63	43.51
	CaFM [30]	44.23	38.55	36.30	45.71	39.92	36.87	50.12	45.87	43.79
	STDO	**45.75**	**40.17**	**38.62**	**46.34**	**41.13**	**38.76**	**50.58**	**46.43**	**44.62**
			game-45s			**inter-45s**			**vlog-45s**	
		×2	×3	×4	×2	×3	×4	×2	×3	×4
ESPCN	awDNN [54]	35.42	30.63	28.65	38.64	31.97	28.32	45.71	41.40	39.20
	NAS [55]	35.55	30.67	28.74	38.81	32.14	28.61	45.81	41.52	39.29
	CaFM [30]	36.09	31.06	29.05	38.88	32.22	28.75	46.19	41.72	39.52
	Ours	**37.75**	**32.29**	**29.96**	**41.20**	**32.48**	**29.09**	**46.33**	**42.26**	**40.26**
SRCNN	awDNN [54]	35.05	30.50	28.59	38.66	31.78	28.25	45.87	41.58	39.29
	NAS [55]	35.15	30.55	28.61	38.79	31.93	28.38	45.95	41.66	39.36
	CaFM [30]	35.49	30.63	28.66	38.88	32.02	28.48	46.18	**41.85**	39.52
	STDO	**36.74**	**31.46**	**29.37**	**41.15**	**32.17**	**28.65**	**46.33**	41.81	**39.69**
VDSR	awDNN [54]	40.29	34.53	31.28	41.99	33.80	30.34	47.61	42.92	40.94
	NAS [55]	41.37	34.92	32.42	42.40	34.53	31.10	47.88	43.33	41.23
	CaFM [30]	41.92	35.56	33.16	42.86	34.49	30.95	48.00	43.50	41.38
	STDO	**42.65**	**36.23**	**33.76**	**43.36**	**35.64**	**31.77**	**48.17**	**43.67**	**41.49**
EDSR	awDNN [54]	42.11	35.75	33.33	42.73	34.49	31.34	47.98	43.58	41.53
	NAS [55]	43.22	36.72	34.32	43.31	35.80	32.67	48.48	44.12	42.12
	CaFM [30]	43.32	37.19	34.61	43.37	35.62	32.35	48.45	44.11	42.16
	STDO	**45.65**	**39.93**	**37.24**	**44.52**	**38.28**	**35.51**	**49.84**	**45.47**	**43.07**
WDSR	awDNN [54]	42.61	36.17	33.85	42.94	34.71	31.81	48.02	44.16	42.19
	NAS [55]	43.72	37.25	34.93	43.41	36.05	33.11	48.52	44.75	42.80
	CaFM [30]	43.97	37.64	35.12	43.52	36.03	32.97	48.51	44.72	42.87
	STDO	**45.71**	**40.33**	**37.76**	**44.54**	**38.72**	**36.03**	**49.76**	**45.95**	**43.99**

of the data sampled. We train a single model by solving the following optimization problem using the joint dataset

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & f_{joint}(\mathbf{w}; \mathcal{D}_{joint}) \\ \text{subject to} \quad & \mathcal{D}_{joint} \in \{\hat{\mathcal{D}}_0, \rho_1 \odot \hat{\mathcal{D}}_1, \dots, \rho_{k-1} \odot \hat{\mathcal{D}}_{k-1}\}, \\ & \textstyle\sum_{i=0}^{k-1} \|\rho_i \odot \hat{\mathcal{D}}_i\| = \mu, \end{aligned} \tag{2}$$

where $\odot$ is the sampling operation with pre-defined proportion ρ, and μ is a hyper-parameter that control the size of the joint dataset.

4. Experimental Results

In this section, we conduct extensive experiments to prove the advantages of our proposed methods. To show the effects of our methods, we apply our proposed STDO and JSTDO to videos with different scenes and different time lengths. The detailed information on video datasets and implementations are shown in Section 4.1. In Section 4.2, we compared our method with time-divided method using different videos and different SR models, which show that STDO achieves outperforming video super-resolution quality as well as using lowest computation budgets. In section 4.3, we demonstrate the results of our single SR model obtained by JSTDO and show that JSTDO effectively exploits heterogeneous information density among different video chunks to achieve better training performance. In Section 4.4, we deploy our model on an off-the-shelf mobile phone to show our model can achieve real-time video super-resolution. In Section 4.5, we show our ablation study on key parameters used in STDO and JSTDO methods, such as the different number of chunks, training data scheduling,

Table 3. Computation cost for different backbones with VSD4K video game-45s. We include the computation cost for the models with different resolution upscaling factors.

Model	Scale	FLOPs	CaFM [30]	STDO
ESPCN	×2	0.14G	36.09	37.75
	×3	0.15G	31.06	32.29
	×4	0.16G	29.05	29.96
SRCNN	×2	0.64G	35.49	36.74
	×3	1.45G	30.63	31.46
	×4	2.58G	28.66	29.37
VDSR	×2	6.15G	41.92	42.65
	×3	13.85G	35.56	36.23
	×4	20.62G	33.16	33.76
EDSR	×2	3.16G	43.32	45.65
	×3	3.60G	37.19	39.93
	×4	4.57G	34.61	37.24
WDSR	×2	2.73G	43.97	45.71
	×3	2.74G	37.64	40.33
	×4	2.76G	35.12	37.76

and long video with multiple scene conversions.

4.1. Datasets and Implementation Details

In the previous video super-resolution works, most video datasets [33, 52] for super-resolution only provide several adjacent frames as a mini-video. Those mini-video sets are not suitable for a network to overfit. Therefore, we adopt the VSD4K collected in [30]. In this video dataset, there are 6 video categories including: vlog, game, interview, city, sports, dance. Each of the category contains various video lengths. We set the resolution for HR videos to 1080p, and LR videos are generated by bicubic interpolation to match different scaling factors.

We apply our approach to several popular SR models including ESPCN [40], SRCNN [10], VDSR [21], EDSR [28], and WDSR [57]. During training, we use patch sizes of 48×54 for scaling factor 2 and 4, and 60×64 for scaling factor 3 to accommodate HR video size, and the threshold value λ is set to split the patches evenly into chunks. Regarding the hyperparameter configuration of training the SR models, we follow the setting of [28,30,57]. We adopt Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.009$, $\epsilon = 10^{-8}$ and we use L1 loss as loss function. For learning rate, we set 10^{-3} for WDSR and 10^{-4} for other models with decay at different training epochs. We conduct our experiment on EDSR model with 16 resblocks and WDSR model with 16 resblocks.

4.2. Compare with the State-of-the-Art Methods

In this section, we compare our method with the state-of-the-art (SOTA) that either use general model overfitting or time-divided model overfitting on different SR backbones. Due to space limits, we sample three video categories from VSD4K, and test on two different video time lengths as $15s$ and $45s$. Our results are shown in Table 2. We compare with the state-of-the-art neural network-based SR video delivery methods, such as awDNN [54] where a video is overfitted by a model, NAS [55] that splits a video into multiple chunks in advance and overfit each of the time-divided chunk with independent SR model, and CaFM [30] that uses time-divided video chunk and single SR model with handcrafted module to overfit videos. For our implementation of STDO, we divide the spatial-temporal data into 2, 4, and 6 chunks respectively, and report the best results. We adjust batch size while training to keep the same computation cost, and we show the comparison by computing the PSNR of each method. It can be seen that our method can exceed the SOTA works consistently on different backbones.

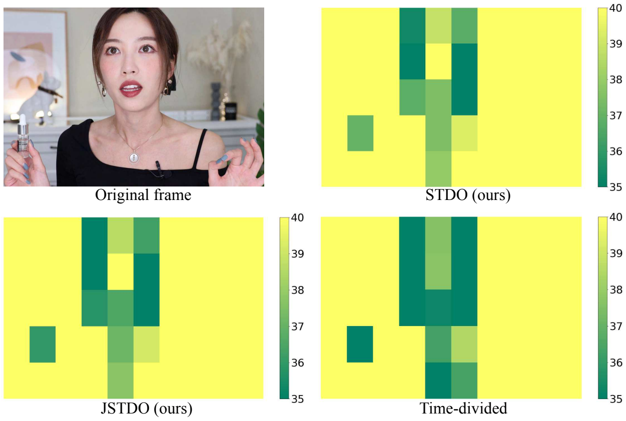

Original frame
STDO (ours)
JSTDO (ours)
Time-divided

Figure 3. PSNR heatmaps of super-resolving an LR video with different methods. STDO and JSTDO has similar value in the key content zone (i.e., body), and outperform time-divided method.

With STDO, each SR model is only overfitting one video chunk that has similar information density, which makes it suitable to use smaller and low capacity SR models that has low computations. In Table 3, we demonstrate the computation cost on each model. From the results, we notice that with the relatively new model such as VDSR, EDSR, and WDSR, when the computation drops below 3 GFLOPs, time-divided method experiences significant quality degradation, while STDO maintains its performance or even achieves quality improvements. When using extremely small networks such as ESPCN or SRCNN, time-divided methods PSNR drops quickly, while STDO still achieves 0.7 ∼ 1.7 dB better performance.

4.3. Data-Aware Joint Training

In this part, we show the results of reducing the number of SR models to a single model by data-aware joint training with the spatial-temporal data overfitting (JSTDO) in

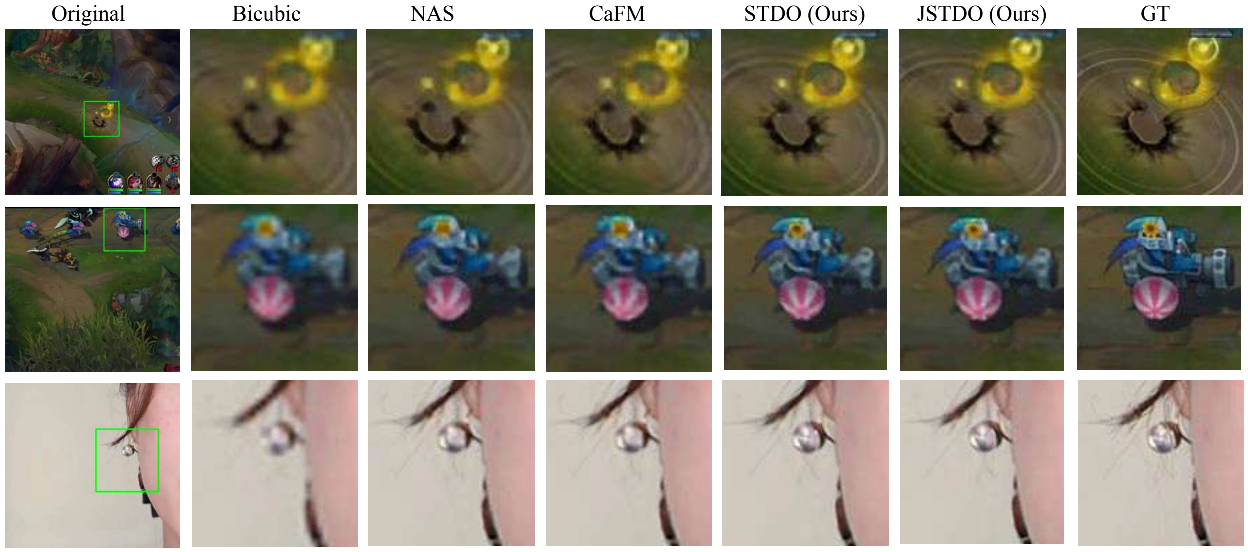


Figure 4. Super-resolution quality comparison with random video frame using STDO and JSTDO with baseline methods.

Table 4. Comparison between STDO and JSTDO regarding PSNR and total number of model parameters with game-15s from VSD4K. We compute the PSNR difference of the two methods.

Model	Method	#Chunks	#Models	#Param.	PSNR
WDSR×2	STDO	4	4	4.8M	45.75
	JSTDO	4	**1**	**1.2M**	45.46
Δ_{PSNR}: 0.29					
WDSR×3	STDO	4	4	4.8M	40.17
	JSTDO	4	**1**	**1.2M**	39.87
Δ_{PSNR}: 0.30					
WDSR×4	STDO	4	4	4.8M	38.62
	JSTDO	4	**1**	**1.2M**	38.14
Δ_{PSNR}: 0.48					

Table 4. We conduct our experiment on the relatively latest model WDSR [57] to chase a better recovering performance. In our experiments, we use 4 chunks for WDSR implementation, and compare the total number of parameters and PSNR of STDO with those of JSTDO which only uses *one* SR model with the same model architecture used by STDO. From the results, we can clearly see that JSTDO effectively reduces the overall model size from 4.8 MB to 1.2 MB, while the PSNR of JSTDO has only negligible degradation compared with STDO. Please note that even with some minor quality degradation, JSTDO still outperforms baseline methods in both super-resolution quality and total parameter counts.

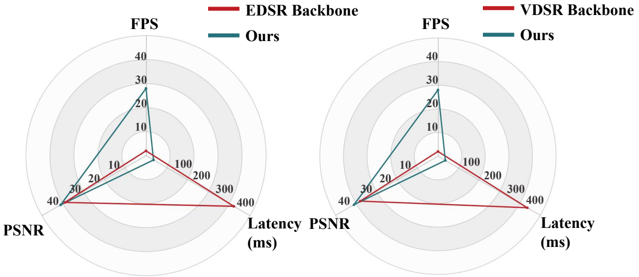


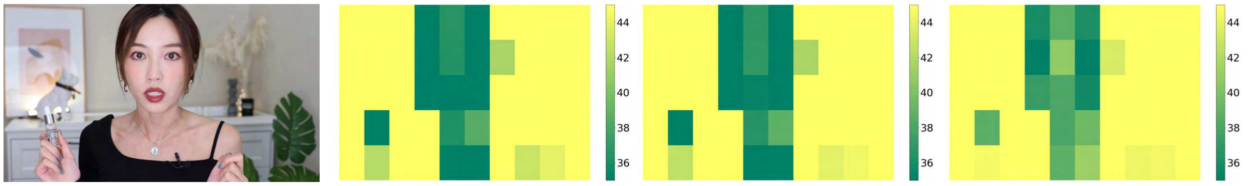
Figure 5. Execution speed and video quality comparison between STDO and [30] [54] respectively using an Samsung mobile phone.

We also plot the PSNR heatmaps when using STDO and JSTDO for video super-resolution, and compare with the traditional time-divided method. As showing in Figure 3, we randomly select one frame in a vlog-15s video that is super-resolved from 270p to 1080p (×4) by CaFM [30], STDO, and JSTDO. The heatmaps clearly demonstrate that our methods achieve better PSNR in the key content zone in the frame. Meanwhile, another key observation can be drawn: the JSTDO heatmap has similar patterns with the one using STDO, which further proves that the joint training technique using carefully scheduled spatial-temporal data effectively captures the key features, while not losing the expressive power towards the low information density data. We also show the qualitative comparison in Figure 4.

4.4. Deployment on Mobile Devices

One of the many benefits by using STDO is that we can use smaller (i.e., low model capacity & complexity) SR models to perform data overfitting. The reason is that the patches in each chunk are relatively similar, especially for some short videos, which makes it easier for smaller models to "memorize" them. Subsequently, unlike CaFM [30], no handcrafted modules are needed for both STDO and JSTDO methods, which further reduces the compilation burden on the end-user devices.

We deploy the video chunks alongside with the overfit-



Original frame image

10% most informative data

50% most informative data

95% most informative data

Figure 6. PSNR heatmap with different ratio of $\hat{\mathcal{D}}_0$ in JSTDO. More informative data participates in joint training achieves better PSNR.

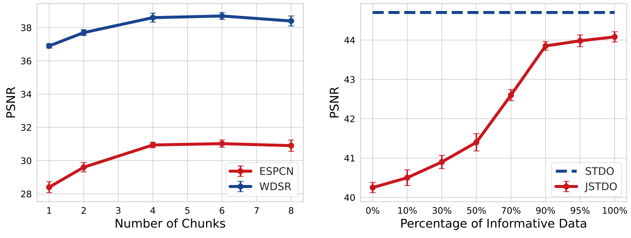
ting models of STDO on a Samsung Galaxy S21 with Snapdragon 888 to test execution performance. Each patch will have a unique index to help assemble into frames. Our results are shown in Figure 5. We set the criteria for real-time as latency less than 50 ms and FPS greater than 20 on the mobile devices according to [59]. The result shows that our method achieves 28 FPS when super-resolving videos from 270p to 1080p by WDSR, and it is significantly faster in speed and better in quality than other models such as EDSR or VDSR that are originally used in other baseline methods [30, 53–55]. Please note that the capability of using small scale SR models to accelerate execution speed is ensured by the high super-resolution quality achieved by spatial-temporal data overfitting method.

4.5. Ablation Study

▷ **Different number of chunks in STDO.** Previous experiments have proved that STDO brings performance improvement when we take account of the spatial-temporal information. In this ablation study, we vary the number of chunks and evaluate their video super-resolution quality. We set the number of chunks with the range of 1 (i.e., single model overfitting) to 8, and we plot the PSNR trends using ESPCN and WDSR in Figure 7a. We observe that ESPCN and WDSR demonstrate similar trends when the number of chunks increases, and better results can be obtained when we divide video into $\sim$4 chunks, which consolidates our claim that STDO uses fewer chunks compared to time-divided methods.

▷ **Data scheduling in joint training.** In JSTDO, we vary the sampling proportion by increasing patches from $\hat{\mathcal{D}}_0$ while decreasing the proportion of patches in $\hat{\mathcal{D}}_k$, and adjusting sampling proportion for other chunks accordingly to maintain the same amount of training patches. The evaluation results are shown in Figure 7b, where we observe that when more informative data participates in training, the overall video super-resolution quality increases. Same patterns can be seen in Figure 6, where the heatmaps show relatively high PSNR in the key content zone when the SR model is trained with more informative data.

▷ **Long video with multiple scene conversions.** We combine the game-45s video and the vlog-45s video together into a 90s long video which contains multiple scene conversions. The STDO result of the long video trained on WDSR is 39.92 dB with scaling factor 4, which is close to the average value of overfitting two videos (43.99 dB and 37.76 dB). Therefore, it can be proved that our design can still maintain good performance for long videos where multiple scene conversions exist.



(a)

(b)

Figure 7. (a) PSNR of video game-15s on different number of chunks, (b) Comparison of different data schedule in joint training.

5. Conclusion

In this paper, we introduce a novel spatial-temporal data overfitting approach towards high-quality and efficient video resolution upscaling tasks at the user end. We leverage the spatial-temporal information based on the content of the video to accurately divide video into chunks, then overfit each video chunk with an independent SR model or use a novel joint training technique to produce a single SR model that overfits all video chunks. We successfully keep the number of chunks and the corresponding SR models to a minimum, as well as obtaining high super-resolution quality with low capacity SR models. We deploy our method on the mobile devices from the end-user and achieve real-time video super-resolution performance.

6. Acknowledgment

This work is partly supported by the National Science Foundation IIS-1949640, CNS-2008049, CNS-2232048, and CNS-2204445, and Air Force Office of Scientific Research FA9550-20-1-0090. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF and AFOSR.

References

[1] Yoshua Bengio, Yann LeCun, et al. Scaling learning algorithms towards ai. *Large-scale kernel machines*, 34(5):1–41, 2007. 2

[2] Jose Caballero, Christian Ledig, Andrew Aitken, Alejandro Acosta, Johannes Totz, Zehan Wang, and Wenzhe Shi. Real-time video super-resolution with spatio-temporal networks and motion compensation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4778–4787, 2017. 3

[3] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12299–12310, 2021. 3

[4] Jiawen Chen, Miao Hu, Zhenxiao Luo, Zelong Wang, and Di Wu. Sr360: boosting 360-degree video streaming with super-resolution. In *Proceedings of the 30th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, pages 1–6, 2020. 2, 3

[5] Tong Chen, Haojie Liu, Qiu Shen, Tao Yue, Xun Cao, and Zhan Ma. Deepcoder: A deep neural network based video compression. In *2017 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4. IEEE, 2017. 1

[6] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 3

[7] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11065–11074, 2019. 3

[8] Mallesham Dasari, Arani Bhattacharya, Santiago Vargas, Pranjal Sahu, Aruna Balasubramanian, and Samir R Das. Streaming 360-degree videos using super-resolution. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pages 1977–1986. IEEE, 2020. 2, 3

[9] Amar B Deshmukh and N Usha Rani. Fractional-grey wolf optimizer-based kernel weighted regression model for multi-view face video super resolution. *International Journal of Machine Learning and Cybernetics*, 10(5):859–877, 2019. 1

[10] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015. 1, 3, 6

[11] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *European conference on computer vision*, pages 391–407. Springer, 2016. 1, 3

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3

[13] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015. 1

[14] Yang Fan, Fei Tian, Tao Qin, Jiang Bian, and Tie-Yan Liu. Learning what data to learn. *arXiv preprint arXiv:1702.08635*, 2017. 2

[15] Hans G Feichtinger and Karlheinz Gröchenig. Iterative reconstruction of multivariate band-limited functions from irregular sampling values. *SIAM journal on mathematical analysis*, 23(1):244–261, 1992. 1

[16] Amirhossein Habibian, Ties van Rozendaal, Jakub M Tomczak, and Taco S Cohen. Video compression with rate-distortion autoencoders. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7033–7042, 2019. 1

[17] Jun Han, Salvator Lombardo, Christopher Schroers, and Stephan Mandt. Deep probabilistic video compression. 2018. 1

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[19] Mehrdad Khani, Vibhaalakshmi Sivaraman, and Mohammad Alizadeh. Efficient video compression via content-adaptive super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4521–4530, 2021. 1, 3

[20] Jaehong Kim, Youngmok Jung, Hyunho Yeo, Juncheol Ye, and Dongsu Han. Neural-enhanced live streaming: Improving live video ingest via online learning. In *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*, pages 107–125, 2020. 1, 2, 3

[21] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1646–1654, 2016. 1, 3, 6

[22] Tae Hyun Kim, Mehdi SM Sajjadi, Michael Hirsch, and Bernhard Scholkopf. Spatio-temporal transformer network for video restoration. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 106–122, 2018. 3

[23] M Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. *Advances in neural information processing systems*, 23, 2010. 2

[24] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017. 1, 3

[25] Royson Lee, Stylianos I Venieris, Lukasz Dudziak, Sourav Bhattacharya, and Nicholas D Lane. Mobisr: Efficient on-device super-resolution through heterogeneous mobile processors. In *The 25th annual international conference on mobile computing and networking*, pages 1–16, 2019. 1

[26] Xiaoqi Li, Jiaming Liu, Shizun Wang, Cheng Lyu, Ming Lu, Yurong Chen, Anbang Yao, Yandong Guo, and Shanghang Zhang. Efficient meta-tuning for content-aware neural video delivery. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVIII*, pages 308–324. Springer, 2022. 3

[27] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1833–1844, 2021. 3

[28] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017. 1, 2, 3, 6

[29] Hongying Liu, Zhubo Ruan, Peng Zhao, Chao Dong, Fanhua Shang, Yuanyuan Liu, Linlin Yang, and Radu Timofte. Video super-resolution based on deep learning: a comprehensive survey. *Artificial Intelligence Review*, pages 1–55, 2022. 3

[30] Jiaming Liu, Ming Lu, Kaixin Chen, Xiaoqi Li, Shizun Wang, Zhaoqing Wang, Enhua Wu, Yurong Chen, Chuang Zhang, and Ming Wu. Overfitting the data: Compact neural video delivery via content-aware feature modulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4631–4640, 2021. 2, 3, 5, 6, 7, 8

[31] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. Dvc: An end-to-end deep video compression framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11006–11015, 2019. 1

[32] Yiqun Mei, Yuchen Fan, and Yuqian Zhou. Image super-resolution with non-local sparse attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3517–3526, 2021. 3

[33] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 6

[34] Ben Niu, Weilei Wen, Wenqi Ren, Xiangde Zhang, Lianping Yang, Shuzhen Wang, Kaihao Zhang, Xiaochun Cao, and Haifeng Shen. Single image super-resolution via a holistic attention network. In *European conference on computer vision*, pages 191–207. Springer, 2020. 3

[35] Cheng Peng, Wei-An Lin, Haofu Liao, Rama Chellappa, and S Kevin Zhou. Saint: spatially aware interpolation network for medical slice synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7750–7759, 2020. 1

[36] Zequn Qin, Pengyi Zhang, Fei Wu, and Xi Li. Fcanet: Frequency channel attention networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 783–792, 2021. 3

[37] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021. 3

[38] Oren Rippel, Sanjay Nair, Carissa Lew, Steve Branson, Alexander G Anderson, and Lubomir Bourdev. Learned video compression. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3454–3463, 2019. 1

[39] Mehdi SM Sajjadi, Raviteja Vemulapalli, and Matthew Brown. Frame-recurrent video super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6626–6634, 2018. 1, 3

[40] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. 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*, pages 1874–1883, 2016. 1, 3, 6

[41] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[42] Xin Tao, Hongyun Gao, Renjie Liao, Jue Wang, and Jiaya Jia. Detail-revealing deep video super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4472–4480, 2017. 3

[43] Yapeng Tian, Yulun Zhang, Yun Fu, and Chenliang Xu. Tdan: Temporally-deformable alignment network for video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3360–3369, 2020. 3

[44] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang. Ntire 2017 challenge on single image super-resolution: Methods and results. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 114–125, 2017. 1

[45] Brian C Tom, Aggelos K Katsaggelos, and Nikolas P Galatsanos. Reconstruction of a high resolution image from registration and restoration of low resolution images. In *Proceedings of 1st international conference on image processing*, volume 3, pages 553–557. IEEE, 1994. 1

[46] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*, 2018. 2, 4

[47] Tong Tong, Gen Li, Xiejie Liu, and Qinquan Gao. Image super-resolution using dense skip connections. In *Proceedings of the IEEE international conference on computer vision*, pages 4799–4807, 2017. 1

[48] Hua Wang, Dewei Su, Chuangchuang Liu, Longcun Jin, Xianfang Sun, and Xinyi Peng. Deformable non-local network for video super-resolution. *IEEE Access*, 7:177734–177744, 2019. 3

[49] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 3

[50] Chao-Yuan Wu, Nayan Singhal, and Philipp Krahenbuhl. Video compression through image interpolation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 416–431, 2018. 1

[51] Xuedou Xiao, Wei Wang, Taobin Chen, Yang Cao, Tao Jiang, and Qian Zhang. Sensor-augmented neural adaptive bitrate video streaming on uavs. *IEEE Transactions on Multimedia*, 22(6):1567–1576, 2019. 2, 3

[52] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8):1106–1125, 2019. 6

[53] Hyunho Yeo, Chan Ju Chong, Youngmok Jung, Juncheol Ye, and Dongsu Han. Nemo: enabling neural-enhanced video streaming on commodity mobile devices. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pages 1–14, 2020. 2, 3, 8

[54] Hyunho Yeo, Sunghyun Do, and Dongsu Han. How will deep learning change internet video delivery? In *Proceedings of the 16th ACM Workshop on Hot Topics in Networks*, pages 57–64, 2017. 1, 4, 5, 6, 7, 8

[55] Hyunho Yeo, Youngmok Jung, Jaehong Kim, Jinwoo Shin, and Dongsu Han. Neural adaptive content-aware internet video delivery. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*, pages 645–661, 2018. 2, 3, 5, 6, 8

[56] Xinyi Ying, Longguang Wang, Yingqian Wang, Weidong Sheng, Wei An, and Yulan Guo. Deformable 3d convolution for video super-resolution. *IEEE Signal Processing Letters*, 27:1500–1504, 2020. 3

[57] Jiahui Yu, Yuchen Fan, Jianchao Yang, Ning Xu, Zhaowen Wang, Xinchao Wang, and Thomas Huang. Wide activation for efficient and accurate image super-resolution. *arXiv preprint arXiv:1808.08718*, 2018. 1, 3, 6, 7

[58] Geng Yuan, Xiaolong Ma, Wei Niu, Zhengang Li, Zhenglun Kong, Ning Liu, Yifan Gong, Zheng Zhan, Chaoyang He, Qing Jin, et al. Mest: Accurate and fast memory-economic sparse training framework on the edge. *Advances in Neural Information Processing Systems*, 34:20838–20850, 2021. 2, 4

[59] Zheng Zhan, Yifan Gong, Pu Zhao, Geng Yuan, Wei Niu, Yushu Wu, Tianyun Zhang, Malith Jayaweera, David Kaeli, Bin Ren, et al. Achieving on-mobile real-time super-resolution with neural architecture and pruning search. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4821–4831, 2021. 8

[60] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 286–301, 2018. 1, 3

[61] Yulun Zhang, Donglai Wei, Can Qin, Huan Wang, Hanspeter Pfister, and Yun Fu. Context reasoning attention network for image super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4278–4287, 2021. 3