COMISR: Compression-Informed Video Super-Resolution

Yinxiao Li, Pengchong Jin, Feng Yang, Ce Liu, Ming-Hsuan Yang, Peyman Milanfar
{yinxiao, pengchong, fengyang, celiu, minghsuan, milanfar}@google.com
Google Inc.

Abstract

Most video super-resolution methods focus on restoring high-resolution video frames from low-resolution videos without taking into account compression. However, most videos on the web or mobile devices are compressed, and the compression can be severe when the bandwidth is limited. In this paper, we propose a new compression-informed video super-resolution model to restore high-resolution content without introducing artifacts caused by compression. The proposed model consists of three modules for video super-resolution: bi-directional recurrent warping, detail-preserving flow estimation, and Laplacian enhancement. All these three modules are used to deal with compression properties such as the location of the intra-frames in the input and smoothness in the output frames. For thorough performance evaluation, we conducted extensive experiments on standard datasets with a wide range of compression rates, covering many real video use cases. We showed that our method not only recovers high-resolution content on uncompressed frames from the widely-used benchmark datasets, but also achieves state-of-the-art performance in super-resolving compressed videos based on numerous quantitative metrics. We also evaluated the proposed method by simulating streaming from YouTube to demonstrate its effectiveness and robustness. The source codes and trained models are available at https://github.com/google-research/google-research/tree/master/comisr.

1. Introduction

Super-resolution is a fundamental research problem in computer vision with numerous applications. It aims to reconstruct detailed high-resolution (HR) image(s) from low-resolution (LR) input(s). When the input is one single image, the reconstruction process usually uses learned image priors to recover high-resolution details of the given image, which is called single-image super-resolution (SISR) [56]. When numerous frames in a video are available, the reconstruction process uses both image priors and inter-frame information to generate temporally smooth high-resolution results, which is known as video super-resolution (VSR).

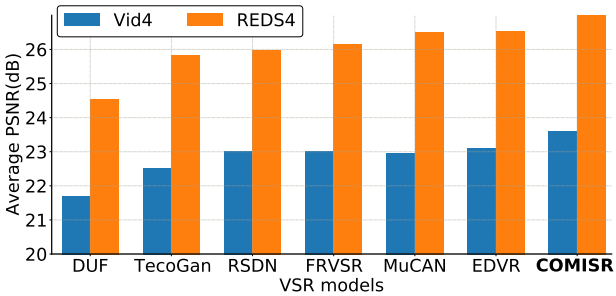


Figure 1. Video super-resolution results (4×, RGB-channels) on compressed Vid4 and REDS datasets. Here we show the results using the most widely adopted compression rate (CRF 23 [10]).

Although great progress has been made, existing SISR and VSR methods rarely take compressed images as input. We note that the *uncompressed* videos used in prior work in fact are high-quality image sequences with low compression rate. As such, these SR methods tend to generate significant artifacts when operating on heavily compressed images or videos. However, most videos on the web or mobile devices are stored and streamed with images compressed at different levels. For example, a wide-used compression rate (Constant Rate Factor (CRF)) for H.264 encoding is 23 as a trade-off between visual quality and file size. We note the state-of-the-art VSR algorithms do not perform well when the input videos are compressed.

To handle compressed videos, one potential solution is to first denoise images and remove compression artifacts in images [35, 36, 58] before applying one of the state-of-the-art VSR models. At first glance, this is appealing since a VSR model is fed with high-quality frames, similar to directly using the evaluation data, such as Vid4 [32]. However, our experiments in Section 4.3 show that this approach would not improve SR results and instead negatively affect the visual quality. With pre-processing, it is likely that the denoising model in the first step will be significantly differ-

ent from the degradation kernel used implicitly during the VSR training process. After the denoising process, the VSR models effectively need to handle more challenging images.

Another possible solution is to train the existing state-of-the-art VSR models on the compressed images. This will enforce the VSR models to account for compression artifacts during the training process. However, our experiments described in Section 4.5 show that simply using compressed frames in model training brings only modest improvement. In fact, without specific changes to the designs of network modules, such training data may even negatively affect the overall performance.

To address the above-mentioned issues, we propose a compression-informed (i.e., compression-aware) super-resolution model that can perform well on real-world videos with different levels of compression. Specifically, we design three modules to robustly restore the missing information caused by video compression. First, a bi-directional recurrent module is developed to reduce the accumulated warping errors from the random locations of the *intra-frame* from compressed video frames [46]. Second, a detail-aware flow estimation module is introduced to recover HR flow from compressed LR frames. Finally, a Laplacian enhancement module is adopted to add high-frequency information to the warped HR frames washed out by video encoding. We refer to this proposed model as *COMpression-Informed video Super-Resolution (COMISR)*.

With the proposed COMISR model, we demonstrate the effectiveness of these modules with ablation studies. We conduct extensive experiments on several VSR benchmark datasets, including Vid4 [32] and REDS4 [41], using videos compressed with different CRF values. We show that the COMISR model achieves significant performance gain on compressed videos (e.g., CRF23), as shown in Figure 1, and meanwhile maintains competitive performance on uncompressed videos. In addition, we present evaluation results based on different combinations of a state-of-the-art VSR model and an off-the-shelf video denoiser. Finally, we validate the robustness of the COMISR model on YouTube videos, which are compressed with proprietary encoders.

The contributions of this paper can be summarized as:

- We introduce a compression-informed model for super-resolving real-world compressed videos and achieve state-of-the-art performance.
- We incorporate three modules that are novel to VSR to effectively improve critical components for video super-resolution on compressed frames.
- We conduct extensive experiments of state-of-the-art VSR models on compressed benchmark datasets. We also present a new setting for evaluating VSR models on YouTube transcoded videos, which is a real-world application scenario that existing evaluation methods do not consider.

2. Related Work

A plethora of super-resolution methods have been developed in the literature based on variational formulations [61] or deep neural networks [1, 56, 62]. In this section, we discuss recent deep models closely related to our work for super-resolution.

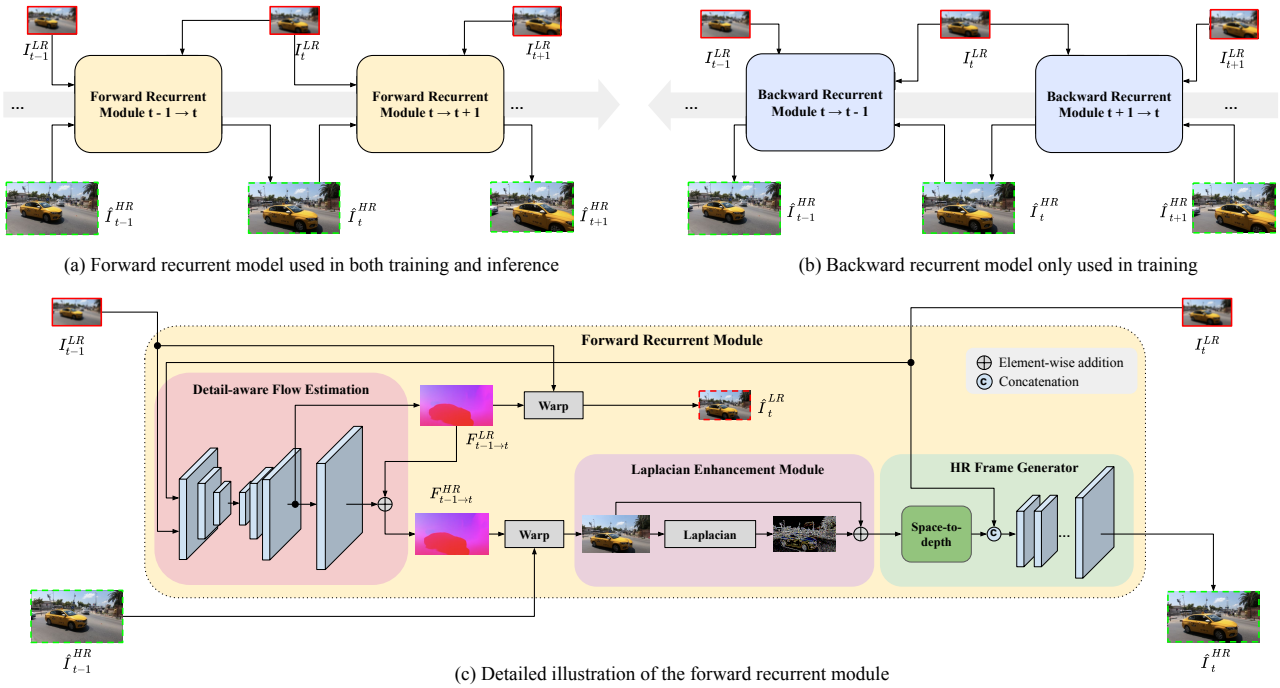
2.1. Single-image Super-resolution

Dong *et al.* [8] propose the SRCNN model based on convolutional neural networks for single image super-resolution. Based on the residual learning framework [18], Kim *et al.* propose the VDSR [24] and DRCN [25] models for more effective image super-resolution. To learn more efficient SR models, Dong *et al*. [9] use a deconvolution layer at the end of the network to directly learn the mapping from low-resolution to high-resolution images. Similarly, Shi *et al*. introduce the ESPCN [47] model with an efficient sub-pixel convolution layer at the end of the network. In the LatticeNet method [38], a light-weighted model is developed by using a lattice block, which reduces half amount of the parameters while maintaining similar SR performance. To learn SR models at multiple scales efficiently, Lai *et al.* [27] develop the LapSRN model which progressively recovers the sub-band residuals of high-resolution images. Instead of relying on deeper models, the MemNet [48] introduce memory block to exploit long-term dependency for effective SR models. On the other hand, the SRDenseNet [50] and RDN [68] are proposed for SISR based on the DenseNet [19] model with dense connections. Haris *et al*. [15] design a deep back-projection network for super-resolution by exploiting iterative up-sampling and down-sampling layers. In [14], the DSRN introduces a dual-state recurrent network model to reduce memory consumption for SISR. The MSRN [29] and RFA [33] models use different blocks to efficiently exploit image features. Recently, attention mechanisms have also been used to improve the super-resolution image quality [5, 40, 42, 67].

Aside from deep neural network models, generative adversarial networks (GANs) have been adopted for SISR, including SRGAN [28], EnhanceNet [44], ESRGAN [55], SPSR [39] and SRFlow [37]. These methods typically generate visual pleasing results by using adversarial losses [12] or normalizing flows [43]. In addition, several models have been developed for SISR based on degrated closer to the real-world scenarios [13, 20, 57, 59, 65].

2.2. Video Super-resolution

Video super-resolution is a more challenging problem than SISR as both content and motion need to be effectively predicted. The motion information provides additional cues in restoring high-resolution frames from multiple low-resolution images.



(a) Forward recurrent model used in both training and inference

(b) Backward recurrent model only used in training

(c) Detailed illustration of the forward recurrent module

Figure 2. Overview of the COMISR model. The forward and backward recurrent modules are symmetric and share the weights. In the figure, red rectangles represent the LR input frames and green dash-lined rectangles represent the HR predicted frames.

Sliding-window methods. Multi-frame super-resolution methods potentially can restore more high-resolution details of target frames as more visual information is available. On the other hand, these methods need to account for motion content between frames for high quality SR results. A number of models compute optical flows between multi-frames to aggregate visual information. Xue *et al*. [60] introduce a task-oriented flow estimation method together with a video processing network for denoising and super-resolution. Haris *et al*. [16] use multiple back-projected features for iterative refinement rather than explicitly aligning frames. Recently, deformable convolution networks [4] have been developed to tackle feature misalignment in dense prediction tasks. Both EDVR [53, 54] and TDAN [49] use deformable convolution models to align features from video frames for video super-resolution. Haris *et al*. [17] design a model that leverages mutually informative relationships between time and space to increase spatial resolution of video frames and interpolate frames to increase the frame rate. In [63], Yi *et al*. propose a model that use non-local blocks to fuse spatial-temporal information from multiple frames. Recenlty, Li *et al*. [30] present a mutli-correspondence network model to exploit spatial and temporal correlation between frames to fuse intra-frame as well as iner-frame information for video SR.

Recurrent models. Recurrent neural networks have been widely used for numerous vision tasks, such as classification [7, 31], detection [34, 51], and segmentation [52]. Such network models can process inputs of any length by sharing model weights across time. In addition, recurrent models can account for long-range dependence among pixels. A number of VSR models have been developed based on recurrent neural networks in recent years. The FRVSR [45] model stores the previous information in a HR frame for restoring the current frame in a sequence. Fuoli [11] use a recurrent latent space to encode and propagate temporal information among frames for video super-resolution. Most recently, the RSDN model [22] incorporates a structure-preserving module into a recurrent network and achieves state-of-the-art performance for restoring details from LR frames without relying on motion compensation.

3. Proposed Method

The COMISR model is designed based on a recurrent formulation. Similar to the state-of-the-art video SR methods [22, 45], it feeds visual information from the previous frames to the current one. The recurrent models usually entail low memory consumption, and can be applied to numerous inference tasks in videos.

Figure 2 shows an overview of the COMISR model. We develop three modules, i.e., bi-directional recurrent warping, detail-aware flow estimation, and Laplacian enhancement modules, to effectively super-resolve compressed videos. Given the LR ground truth frames, we use the forward and backward recurrent modules to generate the HR frame predictions, and compute content losses against HR

ground truth frames in both directions. In the recurrent module, we predict flows and generate warped frames in both LR and HR, and train the network end to end using the LR and HR ground truth frames.

3.1. Bi-directional Recurrent Module

One common approach for video compression is to apply different algorithms to compress and encode frames at different positions in the video stream. Typically, a codec randomly selects several reference frames, known as the *intra-frames*, and compresses them independently without using information from other frames. It then compresses the other frames by exploiting consistency and encoding differences from the *intra-frames*. As a result, the *intra-frames* usually require more bits to encode and have less compression artifacts than the other frames. Since the locations of *intra-frames* is not known in advance, to effectively reduce the accumulated errors from the unknown locations of *intra-frames* for video super-resolution, we propose a bi-directional recurrent network to enforce the forward and backward consistency of the LR warped inputs and HR predicted frames.

Specifically, the bi-directional recurrent network consists of symmetric modules for forward and backward directions. In the forward direction, we first estimate both the LR flow $F^{LR}_{t-1\to t}$ and HR one $F^{HR}_{t-1\to t}$ using the LR frames I^{LR}_{t-1} and I^{LR}_{t} (described in Section 3.2). We then apply different operations separately in LR and HR streams. In the LR stream, we warp the previous LR frame I^{LR}_{t-1} to time t using $F^{LR}_{t-1\to t}$ to obtain the warped LR frame $\tilde{I}^{LR}_{t}$, which will be used at later stages:

$$\tilde{I}^{LR}_{t} = Warp(I^{LR}_{t-1}, F^{LR}_{t-1\to t}). \tag{1}$$

In the HR stream, we warp the previous predicted frames $\hat{I}^{HR}_{t-1}$ to time t using $F^{HR}_{t-1\to t}$ to obtain the warped HR frame $\tilde{I}^{HR}_{t}$, followed by a Laplacian Enhancement Module to generate accurate HR warped frame:

$$\tilde{I}^{HR,Warp}_{t} = Warp(\hat{I}^{HR}_{t-1}, F^{HR}_{t-1\to t}), \tag{2}$$

$$\tilde{I}^{HR}_{t} = Laplacian(\tilde{I}^{HR,Warp}_{t}) + \tilde{I}^{HR,Warp}_{t}. \tag{3}$$

We then apply a space-to-depth operation on $\tilde{I}^{HR}_{t}$ to shrink back its resolution while expanding its channel, fuse it with the LR input I^{LR}_{t} and pass the concatenated frame to the HR frame generator to predict the final HR image $\hat{I}^{HR}_{t}$. We compare $\hat{I}^{HR}_{t}$ with the ground truth HR I^{HR}_{t} to measure the loss.

Similarly, we apply the symmetric operations in the backward direction to obtain the warped LR frame and the predicted HR frame. In this case, the detail-aware flow estimation module generates the backward flow from time t to $t-1$, and images are warped by applying the backward flow to the frame at time t for estimating the frame at time $t-1$.

3.2. Detail-aware Flow Estimation

In our recurrent module, we explicitly estimate both the LR and HR flows between neighboring frames and pass this information in forward and backward directions.

Here we take the forward direction for illustration. The operations in the backward direction are similarly applied. We first concatenate two neighboring LR frames I^{LR}_{t-1} and I^{LR}_{t} and pass it through the LR flow estimation network to estimate the LR flow $F^{LR}_{t-1\to t}$. Instead of directly upsampling the LR flow $F^{LR}_{t-1\to t}$, we add a few additional deconvolution layers on top of the bilinearly upsampled LR flow. Thus, a detailed residual map is learned during the end-to-end training, and we can better preserve high-frequency details in the predicted HR flow.

3.3. Laplacian Enhancement Module

The Laplacian residual has been widely used in numerous vision tasks, including image blending, super-resolution, and restoration. It is particularly useful at finding fine details from a video frame, where such details could be smoothed out during video compression. In our recurrent VSR model, the warped predicted HR frame retains detailed texture information learned from the previous frames. Such details can be easily missing from the up-scaling network, as shown in Figure 2. As such, we add a Laplacian residual to a predicted HR frame to enhance details.

An image is enhanced by Laplacian residuals using a Gaussian kernel blur $G(\cdot,\cdot)$ with the width of σ:

$$\tilde{I}^{HR}_{t} = \tilde{I}^{HR}_{t} + \alpha(\tilde{I}^{HR}_{t} - G(\tilde{I}^{HR}_{t}, \sigma = 1.5)), \tag{4}$$

where $\tilde{I}^{HR}_{t}$ is an intermediate results of the predicted HR frame and α is weighted factor for the residuals. We present more ablation studies in Section 4 to demonstrate the effectiveness of Laplacian residuals for enhancing image details.

By exploiting the Laplacian, we add details back to the warped HR frame. This is followed by a space-to-depth operation, which rearranges blocks of spatial data into depth dimension, and then concatenation with the LR input frame. We pass it through the HR frame generator to obtain the final HR prediction.

3.4. Loss Function

During training, the losses are computed from two streams for HR and LR frames. For loss on HR frames, the $\mathcal{L}_2$ distance is computed between the final outputs and the HR frames. In Section 3.1, we describe our bi-directional recurrent module for improving the model quality. Here, I_t denotes the ground truth frame and $\tilde{I}_t$ denotes the generated frame at time t. For each of the recurrent steps, the predicted HR frames are used to compute losses. The $\mathcal{L}_2$ losses are combined as:

$$\mathcal{L}_{content}^{HR} = \frac{1}{2N}(\underbrace{\sum_{t=1}^{N} ||I_t^{HR} - \hat{I}_t^{HR}||_2}_{\text{forward}} + \underbrace{\sum_{t=N}^{1} ||I_t^{HR} - \hat{I}_t^{HR}||_2}_{\text{backward}}). \tag{5}$$

Each of the warped LR frames from $t-1$ to t is penalized by the $\mathcal{L}_2$ distance with respect to the current LR frame,

$$\mathcal{L}_{warp}^{LR} = \frac{1}{2N}(\underbrace{\sum_{t=1}^{N} ||I_t^{LR} - \tilde{I}_{t-1}^{Warp}||_2}_{\text{forward}} + \underbrace{\sum_{t=N}^{1} ||I_t^{LR} - \tilde{I}_{t-1}^{Warp}||_2}_{\text{backward}}). \tag{6}$$

The total loss is the sum of the HR and LR losses,

$$\mathcal{L}_{total} = \beta \mathcal{L}_{content}^{HR} + \gamma \mathcal{L}_{warp}^{LR}, \tag{7}$$

where β and γ are weights for each loss.

4. Experiments and Analysis

In this section, we first introduce our implementation details and evaluation metrics. We then evaluate our method against the state-of-the-art VSR models on benchmark datasets. In addition, we demonstrate that our method performs better than a baseline method based on a denoiser and a VSR model. We also evaluate the COMISR model on real-world compressed YouTube videos. Finally, we show ablation on the three novel modules with analysis, and user study results.

4.1. Implementation Details

Datasets. We use the REDS [41] and Vimeo [60] datasets for training. The REDS dataset contains more than 200 video sequences for training, each of which has 100 frames with 1280×720 resolution. The Vimeo-90K dataset contains about 65k video sequences for training, each of which has 7 frames with 448×256 resolution. One main difference between these two datasets is the REDS dataset contains images with much larger motion captured from a handheld device. To train and evaluate the COMISR model, the frames are first smoothed by a Gaussian kernel with the width of 1.5 and downsampled by a factor of 4.

We evaluate the COMISR model on the Vid4 [32] and REDS4 [41] datasets (clip# 000, 011, 015, 020). All the testing sequences contain more than 30 frames. In the following experiments, the COMISR model evaluated on the REDS4 dataset is trained with the REDS dataset using the same setting described in [53]. The COMISR model in all the other experiments is trained using the Viemo-90K.

Compression methods. We use the most common setting for the H.264 codec at different compression rates (i.e., different CRF values). The recommended CRF value is between 18 and 28, and the default is 23 (although the CRF value ranges between 0 and 51). In our experiments, we use CRF of 15, 25, and 35 to evaluate video super-resolution with a wide range of compression rates. For fair comparisons, when evaluating other methods, we use the same degradation method to generate the LR sequences before compression. Finally, these compressed LR sequences are fed into the VSR models for inference.

Training process. For each video frame, we randomly crop 128×128 patches from a mini-batch as input. Each mini-batch consists of 16 samples. The α, β, and γ parameters described in Section 3 are set to 1, 20, 1, respectively. The model trained with the loss functions described in the Section 3.4. We use the Adam optimizer [26] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate is set to 5×10^{-5}. While we aim to train the COMISR model for VSR with compressed videos as input, we first feed uncompressed images to the model, and during the last 20% of the training epochs, we randomly add compressed images in the training process with a probability of 50%. The FFmpeg codec is employed for compression with a CRF value randomly selected between 15 and 25. All the models were trained on 8 NVidia Tesla V100 GPUs. More details can be found on the project website.

Evaluation metrics. We use PSNR, SSIM, and LPIPS [66] for quantitative evaluation of video super-resolution results. For the experiments on YouTube videos, we only present video SR results for evaluation since the ground-truth frames are not available.

4.2. Evaluation against the State-of-the-Arts

We evaluate the COMISR model against state-of-the-art VSR methods, including FRVSR [45], DUF [23], EDVR [53], TecoGan [3], MuCAN [30], and RSDN [22]. Three of the evaluated methods are based on recurrent models, whereas the other three use temporal sliding windows (between 5 and 7 frames). When available, we use the original code and trained models, and otherwise implement these methods. For fair comparisons, the LR frames have been generated the same as described in the published work. These LR frames are then compressed and fed into the super-resolution networks for performance evaluation.

For the Vid4 dataset [32], the PSNR and SSIM metrics are measured on both the Y-channel and RGB-channels, as shown in Table 1. We present the averaged performance on uncompressed videos (original sequences), and videos compressed at different levels (CRF15, 25, 35). We also report the individual sequence performance under CRF25. More results on other CRF factors are presented in the supplementary material. Overall, the COMISR method outperforms all the other methods on videos with medium to high compression rates by 0.5-1.0db in terms of PSNR. Meanwhile, our method performs well (2nd or 3rd place) in less compressed videos. Figure 3 shows some results by the evaluated methods from two sequences. The COMISR model can recover more details from the LR frames with fewer compression artifacts. Both quantitative and visual results

Figure 3. Qualitative evaluation on the Vid4 dataset for $4\times$ VSR. The COMISR model can recover more structure details such as faces and boundaries, with much fewer artifacts. Zoom in for best view.

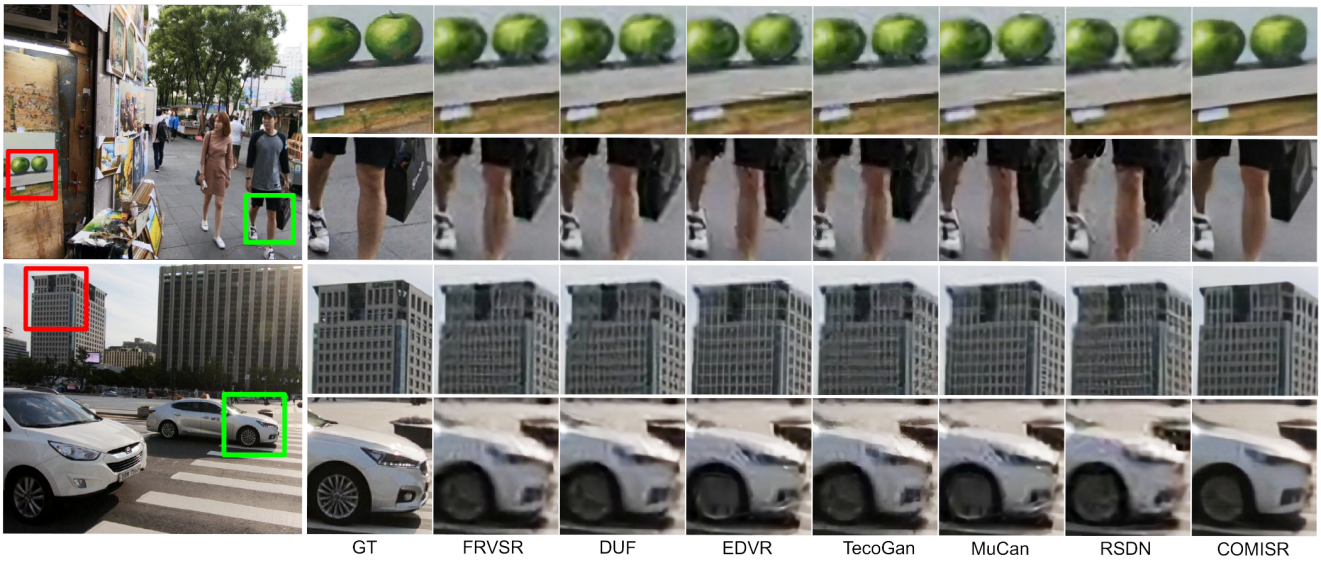


Figure 4. Qualitative results on videos from the REDS4 dataset $4\times$ VSR. The COMISR model achieves much better quality on detailed textures, with much fewer artifacts. The brightness of the images is adjusted for viewing purposes. Zoom in for best view.

show that the COMISR method achieves the state-of-the-art results on compressed videos.

We also evaluate the COMISR model against the state-of-the-art methods on the REDS4 dataset [41]. Unlike the Vid4 dataset, the sequences in this set are longer (100 frames) and more challenging with larger movements between frames. Table 2 shows the COMISR model achieves the best performance on the compressed videos from the REDS4 dataset. Figure 4 shows that our method is able to recover more details such as textures from the bricks on the sidewalk and windows on the buildings.

It is known that low-level structure accuracy (e.g., PSNR or SSIM) does not necessarily correlate well with high-level perceptual quality. In other words, perceptual distortion cannot be well characterized by such low-level structure accuracy [2]. We also use the LPIPS [66] for performance evaluation. Table 3 shows the evaluation results using the LPIPS metric on both Vid4 and REDS4 datasets. Overall, the COMISR model performs well against the state-of-the-art methods on both datasets using the LPIPS metric.

We show video super-resolution results on the project website. Although the compression artifacts are not easily observable in the LR frames, such artifacts are amplified and easily observed after super-resolution. For the compressed videos, the COMISR model effectively recovers more details from the input videos with fewer artifacts.

Model	FLOPs #Param.	CRF 25 calendar	 city	 foliage	 walk	No compression -	Compressed Results CRF15	 CRF25	 CRF35
FRVSR [45]	0.05T 2.53M	21.55 / 0.631 19.75 / 0.606	25.40 / 0.575 23.79 / 0.572	24.11 / 0.625 24.49 / 0.751	26.21 / 0.764 25.22 / 0.815	26.71 / 0.820 25.22 / 0.815	26.01 / 0.766 24.38 / 0.753	24.33 / 0.655 22.59 / 0.640	22.05 / 0.482 20.35 / 0.469
DUF [23]	0.62T 5.82M	21.16 / 0.634 19.40 / 0.588	23.78 / 0.632 22.25 / 0.594	22.97 / 0.603 21.30 / 0.567	24.33 / 0.771 22.66 / 0.737	27.33 / 0.832 25.79 / 0.814	24.40 / 0.773 22.81 / 0.744	23.06 / 0.660 21.41 / 0.621	21.27 / 0.515 19.61 / 0.468
EDVR [53]	0.93T 20.6M	21.69 / 0.648 19.87 / 0.599	25.51 / 0.626 23.90 / 0.586	24.01 / 0.606 22.27 / 0.570	26.72 / 0.786 24.89 / 0.754	27.35 / 0.826 25.85 / 0.808	26.34 / 0.771 24.67 / 0.740	24.45 / 0.667 22.73 / 0.627	22.31 / 0.534 20.62 / 0.487
TecoGan [3]	0.14T 5.05M	21.34 / 0.624 19.55 / 0.601	25.26 / 0.561 23.65 / 0.559	23.50 / 0.592 21.73 / 0.573	25.73 / 0.756 24.40 / 0.743	25.88 / 0.794 24.34 / 0.788	25.25 / 0.741 23.61 / 0.728	23.94 / 0.639 22.22 / 0.624	21.99 / 0.479 20.28 / 0.466
MuCAN [30]	-	21.60 / 0.643 19.81 / 0.597	25.38 / 0.620 23.78 / 0.581	23.93 / 0.599 22.20 / 0.564	26.43 / 0.782 24.72 / 0.750	27.26 / 0.822 25.56 / 0.801	25.85 / 0.753 24.22 / 0.725	24.34 / 0.661 22.63 / 0.623	22.26 / 0.531 20.57 / 0.485
RSDN [22]	0.13T 6.19M	21.72 / 0.650 19.89 / 0.599	25.28 / 0.615 23.68 / 0.575	23.69 / 0.591 21.94 / 0.554	25.57 / 0.747 23.91 / 0.711	**27.92 / 0.851** 26.43 / 0.835	**26.58 / 0.781** 24.88 / 0.750	24.06 / 0.650 22.36 / 0.610	21.29 / 0.483 19.67 / 0.437
COMISR	0.06T 2.63M	**22.81 / 0.695** 20.39 / 0.667	**25.94 / 0.640** 24.30 / 0.633	**24.66 / 0.656** 22.88 / 0.638	**26.95 / 0.799** 25.21 / 0.788	27.31 / 0.840 25.79 / 0.835	26.43 / 0.791 24.76 / 0.778	**24.97 / 0.701** 23.21 / 0.686	**22.35 / 0.509** 20.66 / 0.494

Table 1. Performance evaluation on compressed Vid4 videos. For each entry, the first row is PSNR/SSIM on Y channel, and the second row is PSNR/SSIM on RGB channels. The best method on the Y channel for each column is highlighted in bold and shade. The FLOPs are reported based on the Vid4 $4\times$ VSR. The FLOPs and #Param of FRVSR is based on our implementation.

Model	#Frame	CRF 25 clip_000	 clip_011	 clip_015	 clip_020	No compression -	Compressed Results CRF15	 CRF25	 CRF35
FRVSR [45]	recur(2)	24.25 / 0.631	25.65 / 0.687	28.17 / 0.770	24.79 / 0.694	28.55 / 0.838	27.61 / 0.784	25.72 / 0.696	23.22 / 0.579
DUF [23]	7	23.46 / 0.622	24.02 / 0.686	25.76 / 0.773	23.54 / 0.689	28.63 / 0.825	25.61 / 0.775	24.19 / 0.692	22.17 / 0.588
EDVR [53]	7	24.38 / 0.629	26.01 / 0.702	28.30 / 0.783	25.21 / 0.708	**31.08 / 0.880**	**28.72 / 0.805**	25.98 / 0.706	23.36 / 0.600
TecoGan [3]	recur(2)	24.01 / 0.624	25.39 / 0.682	27.95 / 0.768	24.48 / 0.686	27.63 / 0.815	26.93 / 0.768	25.46 / 0.690	22.95 / 0.589
MuCAN [30]	5	24.39 / 0.628	26.02 / 0.702	28.25 / 0.781	25.17 / 0.707	30.88 / 0.875	28.67 / 0.804	25.96 / 0.705	23.55 / 0.600
RSDN [22]	recur(2)	24.04 / 0.602	25.40 / 0.673	27.93 / 0.766	24.54 / 0.676	29.11 / 0.837	27.66 / 0.768	25.48 / 0.679	23.03 / 0.579
COMISR	recur(2)	**24.76 / 0.660**	**26.54 / 0.722**	**29.14 / 0.805**	**25.44 / 0.724**	29.68 / 0.868	28.40 / 0.809	**26.47 / 0.728**	**23.56 / 0.599**

Table 2. Performance evaluation on compressed the REDS4 dataset. Each entry shows the PSNR/SSIM on RGB channels. The best method for each column is highlighted in bold and shade, and recur(2) indicates a recurrent network using 2 frames.

	FRVSR	TecoGan	DUF	EDVR	MuCAN	RSDN	COMISR
Vid4	4.105	**3.245**	4.010	4.396	3.985	4.292	3.689
REDS4	4.188	3.643	4.223	4.075	4.085	4.423	**3.384**

Table 3. Performance evaluation using the LPIPS [66] metric (lower is better). Our method performs well, especially on the more challenging REDS4 dataset.

The COMISR model does not perform well on highly compressed (e.g., CRF35) videos. Some failure cases are due to heavy compression so that necessary details are missing for super-resolving frames. Other failure cases are caused by extremely large movements in the videos.

4.3. VSR on Denoised Videos

As shown in Figure 3 and Figure 4, the COMISR model generates high-quality frames with fewer artifacts from compressed videos. An interesting question is whether the state-of-the-art methods can achieve better results if the compressed videos are first denoised. As such, we use the state-of-the-art compressed video quality method, STDF [6], for evaluation.

Using the settings described in Section 4.2, we compress video frames with CRF25. The STDF method is then used to remove the compression artifacts and generate enhanced LR frames as inputs for the state-of-the-art VSR methods. Table 4 shows the quantitative results by

Model	VSR only Y-Channel	 RGB-Channels	Video Denoiser + VSR Y-Channel	 RGB-Channels
EDVR	24.45 / 0.667	22.73 / 0.627	22.56 / 0.581	20.94 / 0.541
TecoGan	23.94 / 0.639	22.22 / 0.624	22.25 / 0.541	20.63 / 0.530
MuCan	24.34 / 0.661	22.63 / 0.623	22.47 / 0.577	20.87 / 0.538
RSDN	24.06 / 0.650	22.36 / 0.610	22.19 / 0.560	20.59 / 0.520
COMISR	**24.97 / 0.701**	**23.21 / 0.686**	-	-

Table 4. Ablation study on applying a video denoiser to the compressed frames before the VSR models using the Vid4 dataset. Each entry shows the PSNR/SSIM results on the Y or RGB channel. The COMISR model outperforms the state-of-the-art VSR methods with the STDF [6] denoiser.

the COMISR model and the state-of-the-art VSR methods on videos denosied by the STDF scheme. We note that the performance of all of the evaluated method drops on the denoised LR frames. This can be attributed to that a separate denoising step is not compatible with the learned degradation kernel from the VSR methods. In addition, as discussed in Section 4.5, simply using compressed images for model training does not lead to good VSR performance. These results show that the COMISR model is able to efficiently recover more details from compressed videos, and outperforms state-of-the-art models on denoised videos.

4.4. Evaluation on Real-World Compressed Videos

Most videos on the web are compressed where frames can be preprocessed by a combination of proprietary meth-

Figure 5. 4× VSR results on REDS4 videos downloaded from YouTube with resolution of 360 pixels. Zoom in for best view.

ods. We use the videos from the REDS4 testing dataset for experiments as the image resolution is higher.

We first generate uncompressed videos out of the raw frames, and then upload them to YouTube. These videos are encoded and compressed at different resolutions for downloading. In our setting, the uploaded videos are of 1280 × 720 pixels. The resolutions that are available for downloading on YouTube are 480p, 360p, 240p, and 144p. In the following experiments, we download the videos at 360p using the YouTube-dl [64]. We evaluate three state-of-the-art methods, including MuCAN [30], RSDN [22], and TecoGan [3] on these videos that are compressed by proprietary methods by YouTube. Figure 5 shows the VSR results by the evaluated methods, where the COMISR model produces better visual results with less artifacts.

4.5. Ablation Study

We analyze the contribution of each module in the COMISR model. We start with the recurrent module described in Section 3 as the baseline model. Similar to FRVSR [45], the recurrent model computes the flow between consecutive frames, warps the previous frame to the current, and upscales the frames. We carry out two sets of ablation studies, with or without using compressed images, to show the effectiveness of each module (see Section 4.1).

Table 5 shows the ablation studies where we incrementally add each module to the basic recurrent model. For each setting, the model is trained with and without compressed images, and then evaluated on original and compressed frames. The results show that each module helps achieve additional performance gain, in both training process with only compressed images or a combination of compressed and uncompressed images. We note it is important to add some uncompressed images in the training process to achieve best results on compressed videos. The full COMISR model performs best among all settings. For example, the fourth row in Table 5, the uncompressed PSNR on Vid4 drops 0.17 dB.

4.6. User Study

To better evaluate the visual quality of the generated HR videos, we conduct a user study using Amazon MTurk [21]

Components	No compression Aug		Aug CRF15-25	
	Uncompressed	CRF25	Uncompressed	CRF25
Recur	26.61 / 0.808	23.97 / 0.634	26.53 / 0.815	24.23 / 0.648
Recur + a	27.16 / 0.837	24.24 / 0.650	26.64 / 0.818	24.74 / 0.686
Recur + ab	27.45 / 0.844	24.27 / 0.649	27.27 / 0.838	24.92 / 0.696
Recur + abc	27.48 / 0.845	24.31 / 0.650	27.31 / 0.840	24.97 / 0.701

Table 5. Ablations on three modules of the COMISR model on Vid4: (a) bi-directional recurrent module, (b) detail-aware flow estimation, and (c) Laplacian enhancement module. Each entry shows the PSNR/SSIM values on the Y-channel.

on the Vid4 [32] and REDS4 [41] datasets. We evaluate the COMISR model against all other methods using videos compressed with CRF25. In each experiment, two videos generated by the COMISR model and other methods are presented side by side and each user is asked "which video looks better?" For the Vid4 and REDS4 datasets, all the test videos are used for the user study. For each of the video pairs, we assign to 20 different raters. The aggregated results are shown in Figure 6.

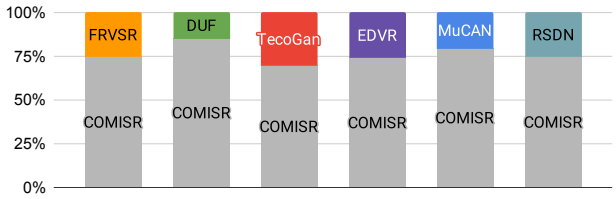


Figure 6. Aggregated user study results on Vid4 and Reds4. Results show that users favored COMISR against all other compared methods.

5. Conclusion

In this work, we propose a compression-informed video super-resolution model which is robust and effective on compressed videos. Within an efficient recurrent network framework, we design three modules to effectively recover more details from the compressed frames. We conduct extensive experiments on challenging video with a wide range of compression factors. The proposed COMISR model achieves the state-of-the-art performance on compressed videos qualitatively and quantitatively, while performing well on uncompressed videos.

References

[1] Saeed Anwar, Salman Khan, and Nick Barnes. A deep journey into super-resolution: A survey. *ACM Computing Surveys*, 2020. 2

[2] Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *CVPR*, 2018. 6

[3] Mengyu Chu, You Xie, Jonas Mayer, Laura Leal-Taixe, and Nils Thuerey. Learning temporal coherence via self-supervision for gan-based video generation. *ACM Transactions on Graphics*, 2018. 5, 7, 8

[4] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017. 3

[5] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *CVPR*, 2019. 2

[6] Jianing Deng, Li Wang, Shiliang Pu, and Cheng Zhuo. Spatio-temporal deformable convolution for compressed video quality enhancement. In *AAAI*, 2020. 7

[7] Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):677–691, 2017. 3

[8] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *ECCV*, 2014. 2

[9] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *ECCV*, 2016. 2

[10] FFmpeg. FFmpeg h.264 video encoding guide. In *https://trac.ffmpeg.org/wiki/Encode/H.264*. 1

[11] Dario Fuoli, Shuhang Gu, and Radu Timofte. Efficient video super-resolution through recurrent latent space propagation. In *ICCV Workshops*, 2019. 3

[12] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 2

[13] Yong Guo, Jian Chen, Jingdong Wang, Qi Chen, Jiezhang Cao, Zeshuai Deng, Yanwu Xu, and Mingkui Tan. Closed-loop matters: Dual regression networks for single image super-resolution. In *CVPR*, 2020. 2

[14] Wei Han, Shiyu Chang, Ding Liu, Mo Yu, Michael Witbrock, and Thomas S. Huang. Image super-resolution via dual-state recurrent networks. In *CVPR*, 2018. 2

[15] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Deep back-projection networks for super-resolution. In *CVPR*, 2018. 2

[16] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita. Recurrent back-projection network for video super-resolution. In *CVPR*, 2019. 3

[17] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita. Space-time-aware multi-resolution video enhancement. In *CVPR*, 2020. 3

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 2

[19] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017. 2

[20] Shady Abu Hussein, Tom Tirer, and Raja Giryes. Correction filter for single image super-resolution: Robustifying off-the-shelf deep super-resolvers. In *CVPR*, 2020. 2

[21] Amazon Inc. Amazon mturk. https://www.mturk.com/, 2021. 8

[22] Takashi Isobe, Xu Jia, Shuhang Gu, Songjiang Li, Shengjin Wang, and Qi Tian. Video super-resolution with recurrent structure-detail network. In *ECCV*, 2020. 3, 5, 7, 8

[23] Younghyun Jo, Seoung Wug Oh, Jaeyeon Kang, and Seon Joo Kim. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *CVPR*, 2018. 5, 7

[24] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, 2016. 2

[25] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *CVPR*, 2016. 2

[26] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[27] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *CVPR*, 2017. 2

[28] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017. 2

[29] Juncheng Li, Faming Fang, Kangfu Mei, and Guixu Zhang. Multi-scale residual network for image super-resolution. In *ECCV*, 2018. 2

[30] Wenbo Li, Xin Tao, Taian Guo, Lu Qi, Jiangbo Lu, and Jiaya Jia. Mucan: Multi-correspondence aggregation network for video super-resolution. In *ECCV*, 2020. 3, 5, 7, 8

[31] Ming Liang and Xiaolin Hu. Recurrent convolutional neural network for object recognition. In *CVPR*, June 2015. 3

[32] Ce Liu and Deqing Sun. A Bayesian approach to adaptive video super resolution. In *CVPR*, 2011. 1, 2, 5, 8

[33] Jie Liu, Wenjie Zhang, Yuting Tang, Jie Tang, and Gangshan Wu. Residual feature aggregation network for image super-resolution. In *CVPR*, 2020. 2

[34] Mason Liu, Menglong Zhu, Marie White, Yinxiao Li, and Dmitry Kalenichenko. Looking fast and slow: Memory-guided mobile video object detection, 2019. 3

[35] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Zhiyong Gao, and Ming-Ting Sun. Deep kalman filtering network for video compression artifact reduction. In *ECCV*, 2018. 1

[36] Guo Lu, Xiaoyun Zhang, Wanli Ouyang, Dong Xu, Li Chen, and Zhiyong Gao. Deep non-local kalman network for video compression artifact reduction. *IEEE Transactions on Image Processing*, 29:1725–1737, 2020. 1

[37] Andreas Lugmayr, Martin Danelljan, Luc Van Gool, and Radu Timofte. Srflow: Learning the super-resolution space with normalizing flow. In *ECCV*, 2020. 2

[38] Xiaotong Luo, Yuan Xie, Yulun Zhang, Yanyun Qu, Cuihua Li, and Yun Fu. Latticenet: Towards lightweight image super-resolution with lattice block. In *ECCV*, 2020. 2

[39] Cheng Ma, Yongming Rao, Yean Cheng, Ce Chen, Jiwen Lu, and Jie Zhou. Structure-preserving super resolution with gradient guidance. In *CVPR*, 2020. 2

[40] Yiqun Mei, Yuchen Fan, Yuqian Zhou, Lichao Huang, Thomas S. Huang, and Honghui Shi. Image super-resolution with cross-scale non-local attention and exhaustive self-exemplars mining. In *CVPR*, 2020. 2

[41] 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 *CVPR Workshops*, 2019. 2, 5, 6, 8

[42] 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 *ECCV*, 2020. 2

[43] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *ICML*, 2015. 2

[44] Mehdi S. M. Sajjadi, Bernhard Scholkopf, and Michael Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. In *ICCV*, 2017. 2

[45] Mehdi S. M. Sajjadi, Raviteja Vemulapalli, and Matthew Brown. Frame-Recurrent Video Super-Resolution. In *CVPR*, 2018. 3, 5, 7, 8

[46] Mike Schuster and Kuldip K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997. 2

[47] Wenzhe Shi, Jose Caballero, Ferenc Huszar, 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 *CVPR*, 2016. 2

[48] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *ICCV*, 2017. 2

[49] Yapeng Tian, Yulun Zhang, Yun Fu, and Chenliang Xu. Tdan: Temporally-deformable alignment network for video super-resolution. In *CVPR*, 2020. 3

[50] Tong Tong, Gen Li, Xiejie Liu, and Qinquan Gao. Image super-resolution using dense skip connections. In *ICCV*, 2017. 2

[51] Vivek Veeriah, Naifan Zhuang, and Guo-Jun Qi. Differential recurrent neural networks for action recognition. In *ICCV*, December 2015. 3

[52] Carles Ventura, Miriam Bellver, Andreu Girbau, Amaia Salvador, Ferran Marques, and Xavier Giro-i Nieto. Rvos: End-to-end recurrent network for video object segmentation. In *CVPR*, June 2019. 3

[53] Xintao Wang, Kelvin C.K. Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *CVPR Workshops*, 2019. 3, 5, 7

[54] Xintao Wang, Ke Yu, Kelvin C.K. Chan, Chao Dong, and Chen Change Loy. Basicsr. https://github.com/xinntao/BasicSR, 2020. 3

[55] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *ECCV Workshops*, 2018. 2

[56] Zhihao Wang, Jian Chen, and Steven C.H. Hoi. Deep learning for image super-resolution: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 1, 2

[57] Pengxu Wei, Ziwei Xie, Hannan Lu, Zongyuan Zhan, Qixiang Ye, Wangmeng Zuo, and Liang Lin. Component divide-and-conquer for real-world image super-resolution. In *ECCV*, 2020. 2

[58] Yi Xu, Longwen Gao, Kai Tian, Shuigeng Zhou, and Huyang Sun. Non-local convlstm for video compression artifact reduction. In *ICCV*, 2019. 1

[59] Yu-Syuan Xu, Shou-Yao Roy Tseng, Yu Tseng, Hsien-Kai Kuo, and Yi-Min Tsai. Unified dynamic convolutional network for super-resolution with variational degradations. In *CVPR*, 2020. 2

[60] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 2019. 3, 5

[61] Chih-Yuan Yang, Chao Ma, and Ming-Hsuan Yang. Single-image super-resolution: A benchmark. In *ECCV*, pages 372–386, 2014. 2

[62] Wenming Yang, Xuechen Zhang, Yapeng Tian, Wei Wang, Jing-Hao Xue, and Qingmin Liao. Deep learning for single image super-resolution: A brief review. *IEEE Transactions on Multimedia*, 2019. 2

[63] Peng Yi, Zhongyuan Wang, Kui Jiang, Junjun Jiang, and Jiayi Ma. Progressive fusion video super-resolution network via exploiting non-local spatio-temporal correlations. In *ICCV*, 2019. 3

[64] Youtube-dl. Youtube-downloader. In *https://github.com/ytdl-org/youtube-dl*, 2021. 8

[65] Kai Zhang, Luc Van Gool, and Radu Timofte. Deep unfolding network for image super-resolution. In *CVPR*, 2020. 2

[66] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 5, 6, 7

[67] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, 2018. 2

[68] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *CVPR*, 2018. 2