

Expanding Synthetic Real-World Degradations for Blind Video Super Resolution

Mehran Jeelani^{1,2,*}, Sadbhawna^{4,*}, Noshaba Cheema^{2,3,5}, Klaus Illgner-Fehns¹,
Philipp Slusallek^{2,5}, and Sunil Jaiswal^{1,†}

¹ K|Lens GmbH, Germany, ² Saarland Informatics Campus, Germany, ³ MPI Informatics, Germany
⁴ IIT Jammu, India, ⁵ German Research Center for Artificial Intelligence (DFKI), Germany



Figure 1. **Results on real-world video.** This figure compares the proposed SRWD-VSR algorithm with the existing state-of-the-art VSR methods ($\times 4$ SR). This work shows how varied random degradations can contribute to learning an effective VSR model, especially for real-world video artifacts. (**Zoom-in for best view**)

Abstract

Video super-resolution (VSR) techniques, especially deep-learning-based algorithms, have drastically improved over the last few years and shown impressive performance on synthetic data. However, their performance on real-world video data suffers because of the complexity of real-world degradations and misaligned video frames. Since obtaining a synthetic dataset consisting of low-resolution (LR) and high-resolution (HR) frames are easier than obtaining real-world LR and HR images, in this paper, we propose synthesizing real-world degradations on synthetic training datasets. The proposed synthetic real-world degradations (SRWD) include a combination of the blur, noise, down-sampling, pixel binning, and image and video compression artifacts. We then propose using a random shuffling-based strategy to simulate these degradations on the training datasets and train a single end-to-end deep neural network (DNN) on the proposed larger variation of realistic synthesized training data. Our quantitative and qualitative

comparative analysis shows that the proposed training strategy using diverse realistic degradations improves the performance by 7.1 % in terms of NRQM compared to RealBasicVSR and by 3.34 % compared to BSRGAN on the VideoLQ dataset. We also introduce a new dataset that contains high-resolution real-world videos that can serve as a common ground for bench-marking.

1. Introduction

Video super-resolution (VSR) has extensive applications such as high-definition television, remote sensing, surveillance systems, etc.. [6, 15, 19]. Since the success of deep learning, the VSR approaches are also massively evolving, and these methods can be divided into two categories [14]: I) Methods using synthetic datasets with pre-defined degradations for training, such as VSRNet [11], FRVSR [20], TecoGAN [5], Deep-Blind-VSR [18], BasicVSR [2], BasicVSR++, [3] etc. II) Methods using real-world degradations, such as RealVSR [25], and RealBasicVSR [4]. The methods in category I synthesize LR images by a pre-defined degradation model (PDM) that consists of a bicubic downsampling, Gaussian blur, and noise. Such a degradation model is inadequate and poses a challenge for real-world VSR. Whereas the methods in category II, such as RealVSR [25], try to overcome this issue by introducing a

This work was partially funded by the German Ministry for Education and Research (BMBF) under the grant PLIMASC and Indo-German Science and Technology Center (IGSTC).

*Both authors contributed equally to the paper

†Corresponding author: sunil.jaiswal@k-lens.de

new real-world dataset using a multi-camera device but it lacks generalizability in diverse degradations. In contrast, RealBasicVSR [4] addresses this issue by adopting a very large DNN and using a second-order degradation model [4] that simulates degradations more than once. We observed that significantly less work [4, 25] had been done toward VSR for real-world videos.

We analyzed the performance of these VSR methods on real-world datasets and have shown them in Figure 1. The performance of the existing algorithm suffers because of the lack of diverse degradations in the training dataset, which is usually a synthetic dataset of LR-HR pairs with a pre-defined degradation or second-order degradation. To overcome these issues of diverse degradations we propose to create a pool of *diverse* synthetic real-world degradations (SRWD), and they are as follows:

Blur degradation: Existing methods [13, 29, 30] use isotropic and anisotropic blur kernels to simulate blur degradations whereas real-world images may have different blur patterns. To have a realistic blur kernel, we propose to extract blur kernels from real-world images using KernelGAN [1], a self-supervised kernel extraction algorithm.

Noise degradation: In addition to Gaussian noise in the RGB space, we propose to add signal-dependent Poisson noise in the sensor RAW space to have a realistic noise distribution in the training datasets.

Downsampling degradation: We use standard downsampling operations, such as bilinear, bicubic, etc., as done in the existing methods [13, 29, 30].

The above degradations are applicable for both image and video modes. Below degradations are specifically for video mode, and they are as follows:

Pixel-binning degradation: In the video mode, the resolution is smaller than in the image mode of the same camera. Pixel binning [10] refers to combining the electrical charges of neighboring pixels to form a superpixel, and due to this conversion, degradation is expected. We simulate this degradation in the RAW space with the help of box filters of different sizes [10].

Compression artifact degradations: Here, we simulate degradation because of compression that arises while saving a video and we apply MPEG compression to simulate this degradation.

In this paper, we do not consider other degradations, such as motion blur, as the test datasets do not have motion blur in them, and we focus on VSR to have a fair comparison with the existing algorithms. However, the proposed SRWD can be extended to include such degradations.

In summary, we propose a realistic degradation model incorporating the major source of variations in real-world data and simulate these degradations on the synthetic training data for VSR. The reason behind creating a diverse pool of SRWD is to match the characteristics of real-world

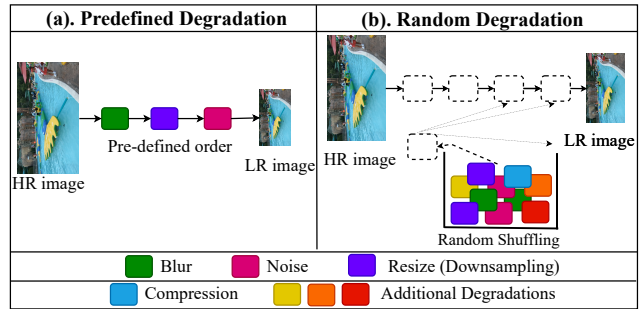


Figure 2. **Synthetic LR-HR pair creation.** Our approach is based on using random degradation techniques, as shown in (b).

videos. To simulate these degradations on the training datasets, we use the random shuffling strategy as proposed in BSRGAN [29]. We then train a VSR deep neural network (DNN) on the proposed synthesized training datasets and perform extensive experiments on completely blind real-world public datasets, such as RealVSR [25] and VideoLQ [4] datasets. The proposed algorithm yields objectively and subjectively superior results compared to the existing real-world VSR methods. We observe that both these public datasets contain low-resolution videos and have camera-specific degradations. Further, to boost the VSR research community, we created a high-resolution real-world VSR dataset captured using the K|Lens [9] and referred to this dataset as the K|Lens dataset.

2. Related Work

Since VSR algorithms trained on a pre-defined degradation model suffer on real-world videos, the RealVSR [25] algorithm addresses this issue by introducing a real-world VSR dataset by capturing the LR-HR video pair from a single device with a multi-camera system (Phone 11 Pro Max). Along with this dataset, the RealVSR [25] provides a benchmark by training different existing PDM-based-VSR algorithms on their proposed dataset. The experimental results and comparison show that these algorithms significantly improve the performance of VSR algorithms. Again, this indicates that learning using a PDM is insufficient and will suffer in real-world videos. However, the main drawback of this benchmark is that it cannot reflect a generalizable benchmark as the RealVSR dataset [25] consists of only degradations specifically for the iPhone camera.

Chan et al. [4] adopt a larger DNN and propose a pre-processing stage to reduce the effect of noise and other degradations before feeding it to the DNN. They analyze how long-term propagation in the training process affects the performance of real-world videos. They proposed using a second-order degradation model [22] that simulates different degradations, such as blur, noise, compression, etc.,

more than once. This is why the degradations are more realistic as compared to the PDM. To validate their performance on real-world datasets, they introduced a new dataset (video) with 50 LR videos obtained using web crawling.

Unlike VSR, several image SR algorithms [28, 30] focus on simulating degradations on the training datasets and achieve impressive performance. Recently, authors in BSRGAN [29] proposed to apply the different degradations in random order instead of following a particular order, as shown in Figure 2(b). The advantage of this strategy is that because of the random order, a different combination of degradations can be simulated, thus expanding the degradation space and training datasets. Inspired by this idea of random shuffling, we follow a similar strategy of simulating diverse degradations in random order. In addition to this random shuffling, this paper focuses on expanding the degradations space by extracting blur kernels from real-world images, signal-dependent sensor noise, pixel-binning, and video compression artifacts. With a diverse pool of SRWD, we then use a random-shuffling-based strategy to create the LR/HR training data. Our experimental results and analysis shows that the random-shuffling strategy-based SRWD can model real-world degradations better than the existing algorithms.

3. Proposed SRWD-VSR Algorithm

In the following subsections, we describe our realistic degradation model (Section 3.1) and VSR architecture (Section 3.2) for our SRWD-VSR algorithm.

3.1. Realistic Degradation Model

The super-resolution process aims at recovering a high-resolution (HR) image from the given low-resolution (LR) image. In general, the LR image (I^{LR}) can be mathematically modeled as,

$$I^{LR} = \Delta(I^{HR}; \delta) \quad (1)$$

where Δ denotes the mapping function of degradation, I^{HR} is the corresponding HR image, and δ represents the various parameters of the degradation process (e.g., blur, downsampling, noise, etc.). Although in real-world images/videos, Δ and δ are unknown, researchers try to imitate the δ parameters such that the degradation Δ is as close as possible to the real-world images. As discussed earlier, also, towards real-world VSR, two approaches have been proposed in the literature. RealVSR [25] tries to get a real I^{HR} and I^{LR} pair by using a multi-camera device, whereas RealBasicVSR [4] trains their model on synthetically created I^{HR} and I^{LR} pairs using a second order degradation model. However, the degradation space is still limited and has the following issues:

1) The blur kernels used in all the existing degradation models are standard isotropic or anisotropic kernels,

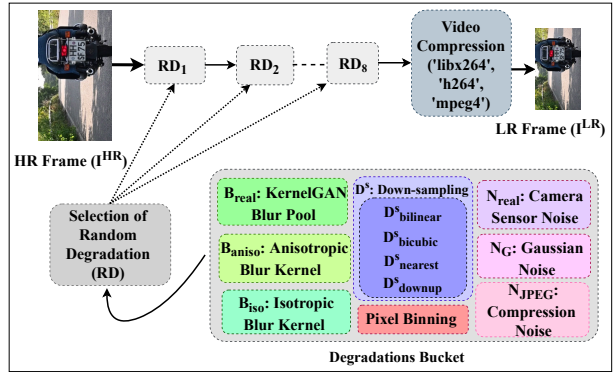


Figure 3. **Proposed SRWD model.** The degradation bucket contains all eight types of degradations from which the random selection is done, which are then followed by video compression.

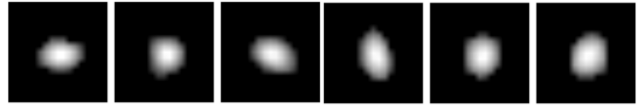


Figure 4. **Kernel Pool.** Some examples from the proposed kernel pool. These kernels are extracted from 6 real-world images using the KernelGAN method [1].

whereas real-world images may have different blur patterns [1]. 2) The real sensor noise is very different from the Gaussian Noise, which is used as a noise factor in these degradation models [27]. 3) Also, not performing any blur operation on RAW space to consider the effect of pixel-binning.

Considering all these points in context with VSR, we propose to expand the degradation space, and we list them below. An overview of the proposed realistic degradation model is shown in Figure 3.

Blur Kernel Pool. Most degradation models rely on fixed blur kernels, such as Gaussian kernel filters [22], to get a synthesized I^{LR} image from a given I^{HR} image. In contrast, real-world LR images do not comply with this assumption. To match the blurring characteristics in real-world images, we propose to extract blur kernels from the real-world images using KernelGAN [1]. The KernelGAN [1] is a dataset invariant, fully unsupervised, and single input image method which can predict the blur kernel in the wild. KernelGAN consists of a generator (G) and a discriminator (D) to downscale. This fully-convolutional G and D work together to predict the SR kernel of the given image. At a patch level, G learns to downscale the given image so that, for D, it is indistinguishable from the image.

In this paper, we create a kernel pool of approximately 5000 kernels from a dataset of 5000 images collected from different sources for generality and we refer to them as (B_{real}). Figure 4 shows some examples of these kernels.

Sensor Noise Addition. Simple noise models like additive white Gaussian noise are not enough for a realistic degradation model as the real-world color images are complicated. Collecting real noisy and clean images is tedious and unsuitable for large-scale learning. We propose injecting realistic noise in the raw sensor space by first converting the image from sRGB to raw space.

Most VSR methods use Gaussian noise and/or a step-by-step technique of inverting camera image signal processing (ISP) to convert sRGB into raw data [29]. The drawback of such an approach is that it needs prior information on the targeting camera device. It is also challenging to reverse several camera imaging pipeline engineering operations. In the proposed degradation model, we use CycleISP [27], an image noise synthesizer that uses a Poisson-Gaussian noise model [7] and can generate realistic synthetic clean/noisy paired data in raw and sRGB spaces. We name the injected noise as N_{real} .

Traditional degradations. Apart from the above-mentioned realistic blur and noise models, adding traditional down-sampling, noise, and blurring can be used for augmentation and make the degradation model more comprehensive [29]. Therefore, the proposed degradation model also uses the following traditional degradations:

Gaussian Blur: In the proposed degradation model, we perform two Gaussian blur operations: B_{iso} with isotropic blur kernels and B_{aniso} with anisotropic blur kernels.

Down-sampling: There are three types of downsampling ways used in the proposed degradation model, i.e., nearest neighbor interpolation ($D_{nearest}^s$), bilinear interpolation ($D_{bilinear}^s$), bicubic interpolation ($D_{bicubic}^s$), down-up sampling ($D_{down-up}^s$), where s is the scaling factor.

Gaussian and JPEG compression noise: We adopted the three-dimensional zero-mean Gaussian noise model (N_G) and JPEG compression noise (N_{JPEG}) with quality factors ranging [30, 95] in the proposed degradation model.

Pixel Binning. In general, blur operations are conducted on sRGB instead of sensor RAW space. Furthermore, in the video mode, the resolution is smaller than in the image mode for the same camera. This decrease in resolution for the same sensor size is achieved by a process commonly referred to as pixel binning [10]. Combining the electrical charges of neighboring pixels to form a super-pixel is termed pixel binning [10]. The main benefit of this technique is that the combined charges would overcome the real noise at the expense of spatial resolution. To simulate the effect of pixel binning, we use box filters of different dimensions for this purpose. The sizes for box filters are randomly chosen to range from 3×3 to 15×15 .

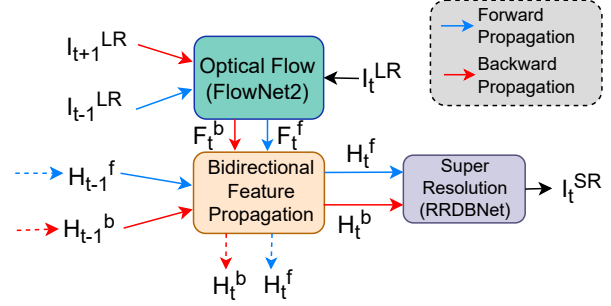


Figure 5. **Proposed VSR Architecture.** On a high level, we calculate the optical flow of the current frame I_t^{LR} with respect to the previous I_{t-1}^{LR} and the next I_{t+1}^{LR} frames. We warp the features H_{t-1}^f and H_{t-1}^b based on the flow calculated (F_t^f and F_t^b) in the Bidirectional Feature Propagation Module and pass the outputs H_t^f and H_t^b to the super-resolution network. The features H_t^f and H_t^b are channel-wise concatenated and fed to the RRDBNet for upsampling.

Video Compression. Once all the degradations are simulated, we apply video compression to LR frames to generate artifacts due to compression. Unlike image degradations, video compression implicitly considers the interdependencies between video frames, providing us with temporally and spatially varying degradations [4]. Therefore, in each iteration, we randomly selected one of the following codecs: “libx264”, “h264”, “mpeg4”, and a bitrate ranging between $[10^4, 10^5]$.

3.1.1 Random Shuffling Strategy

In this paper, we simulate the degradations mentioned above in a random-shuffling strategy as proposed in BSRGAN [29] that helps to expand the degradation space and is depicted in Figure 3. As shown in Figure 3, the proposed SRWD pool contains **1)** Blur Kernels consisting of real-world kernels B_{real} as well as isotropic (B_{iso}) and anisotropic (B_{aniso}) kernels, **2)** Downsampling operations (D^s), **3)** Noise (N_{real} , N_G , N_{JPEG}), **4)** Pixel binning and **5)** video compression.

For a given clean HR video frame (I^{HR}) with no degradations, the proposed algorithm shuffles these eight degradations in a random order, followed by video compression, and applies each of them to generate a realistic synthesized I^{LR} . More specifically, these 8 degradations are : B_{real} , B_{iso} , B_{aniso} , N_{real} , N_G , N_{JPEG} , D^s and pixel binning.

3.2. Video Super Resolution Model

The proposed SRWD model is utilized to simulate a training dataset for a blind VSR and can be used to enhance the performance of any video super-resolution algorithm. For experiments and analysis, in this paper, we adopt the widely-used FRVSR model [20] as a backbone architecture,

with some changes to account for varied, realistic degradation and shuffling strategies. Since the degradations are complex and complicated, it is very vital to have the right sub-module in the architecture and this is why we propose to optimize each of the components in order to have a better super-resolution. In the following, we describe the architecture used for training shown in Figure 5, and the important blocks of the architecture are described below.

Optical Flow. To consider severe degradations, we replace the optical flow estimation module with FlowNet2 [8], which is robust to degradations, such as noise and blur.

Bidirectional Flow. The bidirectional propagation has proven to help learn the VSR better than unidirectional [2]. Hence, the unidirectional propagation of FRVSR is replaced by bidirectional propagation, i.e., each frame receives accumulated information from both directions (forward and backward).

Feature Warping. The image space warping is also replaced with the feature space warping. The reason behind this is that in the image space, the warped images inevitably suffer from blurriness and incorrectness due to the inaccuracy of optical flow estimation [2].

Super resolution. The upsampling module is replaced by RRDBNet [23, 26] to enhance the performance of FRVSR architecture. The first convolutional block of RRDBNet has also been changed to take 128 channel frames as input instead of 3 channel frames because of the introduction of the feature warping module. We replace the VGG-based discriminator with a U-Net-like discriminator with skip connections [22]. The normalized spectral regularization stabilizes the training dynamics, and this architecture can also provide per-pixel feedback to the generator.

We then train the above VSR architecture on the simulated training datasets using loss functions in [22], which is a weighted combination of content loss, perceptual loss, and vanilla adversarial loss.

4. Proposed K|Lens datasets

RealVSR [25], and VideoLQ [4] are two publicly available blind VSR datasets in the literature. The RealVSR testing dataset consists of 50 videos, 50 frames each video, 512×1024 dimension each frame. The VideoLQ dataset also contains 50 videos downloaded from various video-hosting websites. The videos are approximately 100 frames each and have 640×480 resolution for each frame. The videos of the RealVSR dataset are captured using an iPhone camera and hence contain only phone camera-specific degradations, and both these datasets have low-resolution videos.

To further boost the VSR community and to have a high-resolution video dataset, in this paper, we propose introducing a new dataset that contains real-world videos captured using K|Lens [9] camera. The unique optical lens developed

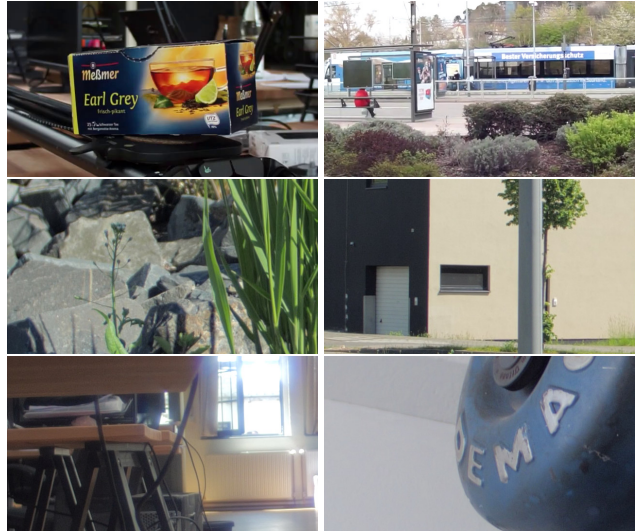


Figure 6. Example frames from the proposed K|Lens dataset. These videos are taken from K|Lens camera with varied scenes (indoor/outdoor) and object motions.

by K|Lens [9] enables any camera with exchangeable lenses to capture multiple perspectives of a scene with a single exposure as regular color images on the camera sensor. It is a new optical lens, and the images or videos shot by K|Lens contain 9 perspectives of a scene, and most importantly, we observe that the number of degradations, such as noise and blur, also vary among these nine perspectives. Thus, shooting one scene results in 9 videos and each of these 9 videos differs in perspectives and amount of degradations. This is one of the reasons why we chose to create a video-SR dataset with K|Lens to have diverse levels of degradations in the datasets. The proposed K|Lens dataset contains 56 videos with 100 frames each with resolution 1600×1080 resolution videos. The dataset consists of indoor and outdoor scenes with varied motions, degradations, and lighting conditions. We have shown a few example scene shots with K|Lens in Fig. 6. Compared to existing datasets, the proposed K|Lens dataset has high-resolution videos with diverse degradation levels.

5. Experiments and Analysis

In this section, we extensively perform experiments on public datasets, such as VideoLQ [4] and RealVSR [25], and also show comparisons on the proposed K|Lens datasets. We show quantitative and quality comparisons in our experiments for analysis and discussion.

5.1. Training

Training datasets Our training dataset combines the Vimeo [24] and REDS [21] datasets. Vimeo consists of 277 video scenes, each having 120 frames, and REDS consists of 240 HD scenes, each having 100 frames. We ex-

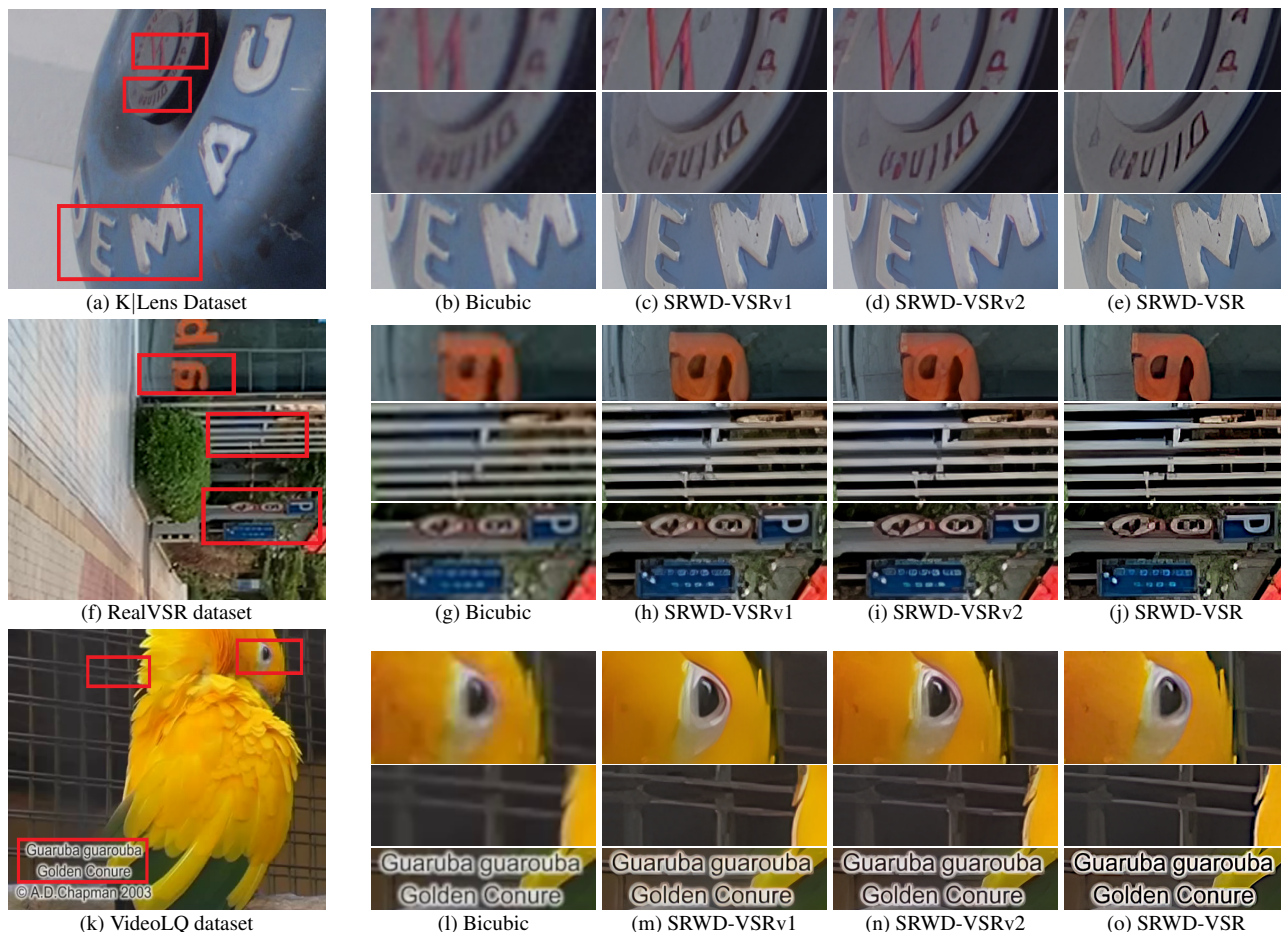


Figure 7. **Qualitative Comparison for all ablations.** These zoomed-in scenes are two times super-resolved images using all the versions of the proposed SRWD-VSR. It can be noticed that the overall results of SRWD-VSR, i.e., (e), (j), and (o), are visibly more plausible. the letter ‘a’ in (j), all the text in (o), and the eye of the bird with much fewer artifacts. (**Zoom-in for best view**)

tract patches of size 128×128 for scale 2 (2 times SR) and 256×256 for scale 4 from these datasets to generate HR videos. We consider these HR datasets and apply the proposed SRDW model to produce the corresponding LR videos. We use 10 consecutive frames of a video and use a batch size of 4, so a total of 40 images are processed in a batch. For degradation simulations with a batch size of 4, each having 10 frames, the first 10 frames will be blurred with the same kernel, same downsampling operations, and have the same noise level setting in addition to pixel-binning and video compression, and the second group of 10 frames may have different blur kernels, noise, etc. from the first group. This is done to avoid a case where subsequent frames with two extreme levels of blur, i.e., no blur and heavy blur, which is not a realistic scenario. Data augmentation consisting of random horizontal and vertical flips is also applied for better generalization.

Training details We implement the network in PyTorch and leverage the built-in Adam optimizer [12] with $\beta_1 = 0.9$ and $\beta_2 = 0.99$ and an initial learning rate of 5×10^{-5} . To

speed up the training, we freeze the weights of Flownet2 [8] as it is already pre-trained and initialize the RRDB network with the model [29] and train the network for 75,000 iterations. Similar to this [22], all degradation operations are implemented with CUDA acceleration to synthesize training pairs on the fly. A training queue is maintained to increase the degradation diversity in the batch. The queue size is set to 180.

5.2. Results and comparison

Here, we adopted three real-world datasets: RealVSR [25], VideoLQ and [4], proposed K|Lens [9] for our experimental analysis. All these three datasets are real-world datasets without ground truths. Therefore, similar to Real-BasicVSR [4], no-reference quality assessment metrics are adopted as the comparison method in this paper. We used NRQM [16] and BRISQUE [17] metrics for this purpose. The higher the value of NRQM better is the performance of the VSR algorithm, while the lower the value of BRISQUE

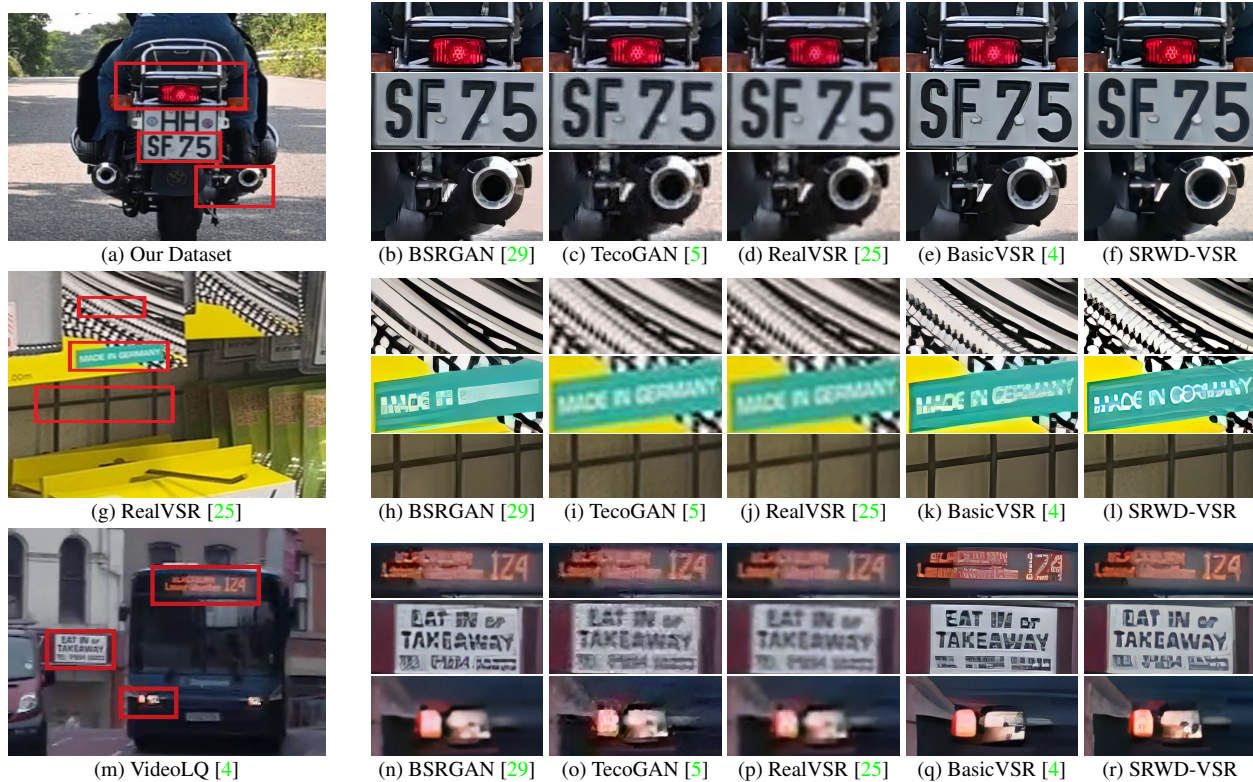


Figure 8. **Comparison with SR Algorithms.** We compared SRWD-VSR with four SOTA algorithms i.e., [4, 5, 25, 29] ($\times 4$ SR). Overall, the SRWD-VSR performs better visibly, for example, ‘Made in Germany’ text in (l) and ‘124’ in (r). (**Zoom in for best view**)

Table 1. **Step-wise performance of our VSR algorithm.** Symbols ‘ \uparrow ’ indicates a higher value is better while ‘ \downarrow ’ indicates a lower value is better. Overall, SRWD-VSR performs consistently better. (Note: These results are for $\times 2$ super-resolution)

	Algorithm	NRQM \uparrow	BRISQUE \downarrow
K Lens	SRWD-VSR	7.1013	26.9868
	SRWD-VSRv2	6.8011	27.3214
	SRWD-VSRv1	6.5914	27.9245
	Bicubic	3.3132	48.9655
RealVSR	SRWD-VSR	6.7507	29.1829
	SRWD-VSRv2	6.4121	32.8701
	SRWD-VSRv1	6.2120	34.9144
	Bicubic	3.3715	55.8189
VideoLQ	SRWD-VSR	6.9054	29.4897
	SRWD-VSRv2	6.8012	30.3145
	SRWD-VSRv1	6.7111	31.9148
	Bicubic	4.0072	45.0282

Table 2. **Degradation analysis of our VSR algorithm.** Symbols ‘ \uparrow ’ indicates a higher value is better while ‘ \downarrow ’ indicates a lower value is better. Overall, SRWD-VSR performs consistently better. (Note: These results are for $\times 2$ super-resolution)

	Algorithm	NRQM \uparrow	BRISQUE \downarrow
K Lens	SRWD-VSR	7.1013	26.9868
	SRWD-VSR-KG	6.9201	27.0948
	SRWD-VSR-SN	6.8014	27.1512
	SRWD-VSR-PB	6.8812	27.2002
RealVSR	SRWD-VSR	6.7507	29.1829
	SRWD-VSR-KG	6.5480	29.9901
	SRWD-VSR-SN	6.5210	30.0124
	SRWD-VSR-PB	6.5912	29.9189
VideoLQ	SRWD-VSR	6.9054	29.4897
	SRWD-VSR-KG	6.8112	29.6814
	SRWD-VSR-SN	6.7614	29.6610
	SRWD-VSR-PB	6.7514	29.6150

better is the VSR algorithm.

To demonstrate the effect of the proposed SRWD model and modified architecture, we train the proposed algorithm with different settings:

SRWD-VSRv1: This version is trained with the pro-

posed SRWD model but without architecture modification and no shuffling.

SRWD-VSRv2: This version is trained with the proposed SRWD model, with proposed architecture modification but with no shuffling.

Table 3. **Performance comparison of proposed SRWD-VSR with existing algorithms.** ‘-’ symbol indicates the unavailability of the pre-trained weights or the training script for that method. It can be observed that the proposed SRWD-VSR performs better in terms of NRQM and BRISQUE parameter values.

		×2 SR		×4 SR	
		NRQM↑	BRISQUE↓	NRQM↑	BRISQUE↓
RealVSR [25]	Proposed	6.7507	29.1829	6.7257	17.5603
	RBVSR [4]	-	-	6.4080	23.6932
	RVSR [25]	3.6927	46.6572	2.8689	51.7278
	Teco [5]	6.7251	38.1057	3.0610	50.1514
	BSR [5]	5.1487	37.3136	5.7911	26.3568
	RESR [22]	5.8664	30.4356	6.0012	28.1113
VideoLQ [4]	Proposed	6.9054	29.4897	6.5993	23.3656
	RBVSR [4]	-	-	6.1618	27.5545
	RVSR [25]	3.3715	31.9713	3.0727	43.7101
	Teco [5]	6.1405	30.9813	5.2951	32.7901
	BSR [29]	6.3892	30.4867	6.3858	30.4369
	RESR [22]	6.3690	27.7421	5.8140	30.5699
K Lens Dataset	Proposed	7.1013	26.9868	6.4678	24.5906
	RBVSR [4]	-	-	6.1259	24.9979
	RVSR [25]	3.3510	40.5177	5.7346	47.9872
	BSR [29]	5.3301	33.3556	5.1832	33.0617
	RESR [22]	5.0024	34.3215	4.8346	35.5532

SRWD-VSR: In this version, shuffling is performed on the proposed SRWD model with the architecture modifications.

Table 1 summarizes the performance of all these versions on three datasets. It can be observed from the table that the proposed SRWD-VSR is better in all the datasets. We have also shown the visual comparisons between these versions to demonstrate the qualitative analysis in Figure 7. It can be observed that the results of SRWD-VSR are more visibly plausible than the compared versions. For instance, only SRWD-VSR could generate patches with negligible artifacts (see Figure 7 (e), (j), (o)). The above experiments give an analysis of the contribution of each of the SRWD models, shuffling-based strategy, and architecture modifications on the proposed SRWD-VSR. This analysis also quantifies the performance gains coming from the architecture modifications as compared to the ones coming from the proposed degradation pipeline. Next, we study the effect of the different degradations in the SRWD model and train the proposed algorithm with different settings as described below:

SRWD-VSR-KG: In this version, we analyze the influence of KernelGAN over isotropic and anisotropic kernels. Instead of generating kernels from KernelGAN, we generate 5000 kernels of isotropic and anisotropic kernels by tuning the variance parameters. In this version, all the proposed degradations are taken into account except 5000 real-world kernels are replaced with 5000 isotropic and

anisotropic kernels. Here, we use the proposed architecture with shuffling-based degradations simulation.

SRWD-VSR-SN: In this version, we analyze the influence of sensor noise. Here, all the degradations are taken into account except sensor noise and we use the proposed architecture modifications and shuffling.

SRWD-VSR-PB: In this version, we discuss the influence of pixel-binning and video compression operations. Here, again all the degradations are taken into account except pixel-binning and video compression degradations and we use proposed architecture modifications and shuffling.

Table 2 summarizes the performance of all these versions on three datasets. Here, SRWD-VSR is the case where all the degradations are taken into account with shuffling and architecture modification. This experiment quantifies the performance gains coming from the different degradations and the removal of any one of the degradations will reduce the performance of the proposed SRWD-VSR algorithm.

Comparison to State-of-the-Art (SOTA) We compare our SRWD-VSR method with several state-of-the-art methods, including RealBasicVSR [4], RealVSR [25], TecoGAN [5], BSRGAN [29], Real-ESRGAN [22]. Since TecoGAN [5] uses the FRVSR [20] generator with an additional discriminator and trains its algorithm with sophisticated loss functions, thus we include TecoGAN, not FRVSR, in the experimental sections and analysis.

We compare these existing state-of-the-art methods subjectively and objectively on scenes from all three datasets, and the results are shown in Table 3 and 8. The proposed algorithm generates much more detail in fine regions, improving visual quality, as seen in Figure 8. Regarding comparison using no-reference performance parameters, i.e., NRQM and BRISQUE, the proposed SRWD-VSR performs better than all the five super-resolution methods for all three tested datasets and is shown in Table 3. Thus, we can conclude that the proposed algorithm is better in all the datasets.

6. Conclusion

LR-HR pairs’ formation for VSR algorithms’ training plays an important role, especially for real-world videos. In this paper, we have investigated a realistic degradation model for video super-resolution methods by introducing a synthetic dataset of real-world degradations and using a random shuffling strategy to train a state-of-the-art VSR DNN architecture. We observed that none of the existing degradation models could imitate the complicated real-world degradations in its LR-HR pairs for training their VSR models to the same extent as ours. The experimental analysis conducted in this paper suggests that the proposed SRWD-VSR performs qualitatively and quantitatively better than all the existing VSR models in the literature.

References

- [1] S. Bell-Kligler, A. Shocher, and M. Irani. Blind super-resolution kernel estimation using an internal-gan. CoRR, Red Hook, NY, USA, 2019. [2](#), [3](#)
- [2] K. Chan, X. Wang, K. Yu, C. Dong, and C. Loy. Basicvsr: The search for essential components in video super-resolution and beyond. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2021. [1](#), [5](#)
- [3] K. Chan, S. Zhou, X. Xu, and C. Loy. BasicVSR++: Improving video super-resolution with enhanced propagation and alignment. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022. [1](#)
- [4] K. Chan, S. Zhou, X. Xu, and C. C. Loy. Investigating trade-offs in real-world video super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022. [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#)
- [5] M. Chu, Y. Xie, M. Jonas, L. Laura, and N. Thuerey. Learning temporal coherence via self-supervision for gan-based video generation (tecogan). *ACM Transactions on Graphics (TOG)*, 39(4), 2020. [1](#), [7](#), [8](#)
- [6] A. Deshmukh and N. 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, 2019. [1](#)
- [7] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE Transactions on Image Processing*, 17(10):1737–1754, 2008. [4](#)
- [8] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. [5](#), [6](#)
- [9] K. Illgner, J. Restrepo, S. P. Jaiswal, and I. Ihrke. Lightfield imaging for industrial applications. In Masafumi Kimata, Joseph A. Shaw, and Christopher R. Valenta, editors, *SPIE Future Sensing Technologies*, volume 11525, page 1152526. International Society for Optics and Photonics, SPIE, 2020. [2](#), [5](#), [6](#)
- [10] X. Jin and K. Hirakawa. Analysis and processing of pixel binning for color image sensor. *EURASIP Journal on Advances in Signal Processing*, 2012:1–15, 2012. [2](#), [4](#)
- [11] A. Kappeler, S. Yoo, Q. Dai, and A. Katsaggelos. Video super-resolution with convolutional neural networks. *IEEE Transactions on Computational Imaging*, 2(2):109–122, 2016. [1](#)
- [12] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015. [6](#)
- [13] C. Liu and D. Sun. On bayesian adaptive video super resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(2):346–360, 2014. [2](#)
- [14] Ruan Z. Zhao P. et al. Liu, H. Video super-resolution based on deep learning: a comprehensive survey. *Springer Artificial Intelligence Review*. [1](#)
- [15] Y. Luo, L. Zhou, S. Wang, and Z. Wang. Video satellite imagery super resolution via convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters*, 14(12):2398–2402, 2017. [1](#)
- [16] C. Ma, C. Yang, X. Yang, and M. Yang. Learning a no-reference quality metric for single-image super-resolution. *Computer Vision and Image Understanding*, 12 2016. [6](#)
- [17] A. Mittal, A. K. Moorthy, and A. C. Bovik. Blind/referenceless image spatial quality evaluator. In *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, pages 723–727, 2011. [6](#)
- [18] J. Pan, H. Bai, J. Dong, J. Zhang, and J. Tang. Deep blind video super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4811–4820, 2021. [1](#)
- [19] C. Peng, W. Lin, H. Liao, R. Chellappa, and S. K. Zhou. Saint: Spatially aware interpolation network for medical slice synthesis. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. [1](#)
- [20] M. Sajjadi, R. Vemulapalli, and M. Brown. Frame-Recurrent Video Super-Resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. [1](#), [4](#), [8](#)
- [21] Sanghyun Son, Suyoung Lee, Seungjun Nah, Radu Timofte, and Kyoung Mu Lee. Ntire 2021 challenge on video super-resolution. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 166–181, 2021. [5](#)
- [22] X. Wang, L. Xie, C. Dong, and Y. Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *International Conference on Computer Vision Workshops (ICCVW)*. [2](#), [3](#), [5](#), [6](#), [8](#)
- [23] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. C. Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *The European Conference on Computer Vision Workshops (ECCVW)*, September 2018. [5](#)
- [24] Tianfan Xue, Baian Chen, Jiajun Wu, D. Wei, and William T. Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, pages 1–20, 2018. [5](#)
- [25] X. Yang, W. Xiang, H. Zeng, and L. Zhang. Real-world video super-resolution: A benchmark dataset and a decomposition based learning scheme. *IEEE Conference on Computer Vision*, 2021. [1](#), [2](#), [3](#), [5](#), [6](#), [7](#), [8](#)
- [26] W. Yifan, F. Perazzi, B. McWilliams, A. Sorkine-Hornung, O. Sorkine-Hornung, and C. Schroers. A fully progressive approach to single-image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. [5](#)
- [27] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M. Yang, and L. Shao. Cycleisp: Real image restoration via improved data synthesis. In *IEEE Computer Vision and Pattern Recognition*, 2020. [3](#), [4](#)
- [28] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, and R. Timofte. Plug-and-play image restoration with deep denoiser prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. [3](#)
- [29] K. Zhang, J. Liang, L. Van Gool, and R. Timofte. Designing a practical degradation model for deep blind image super-

resolution. In *IEEE International Conference on Computer Vision*, pages 4791–4800, 2021. [2](#), [3](#), [4](#), [6](#), [7](#), [8](#)

- [30] K. Zhang, W. Zuo, and L. Zhang. Learning a single convolutional super-resolution network for multiple degradations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3262–3271, 2018. [2](#), [3](#)