# Real-world Video Super-resolution: A Benchmark Dataset and A Decomposition based Learning Scheme

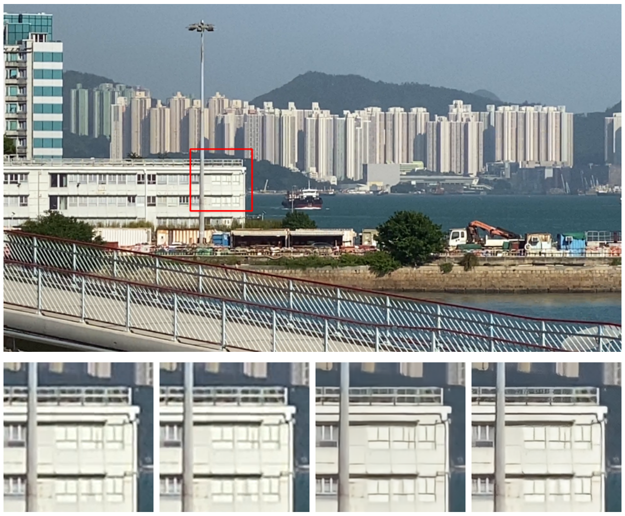
Xi Yang[1,2,*], Wangmeng Xiang[1,2,*], Hui Zeng[1,2], Lei Zhang[1,2,†]
[1]The Hong Kong Polytechnic University, [2]DAMO Academy, Alibaba Group
xxxxi.yang@connect.polyu.hk, {cswxiang, cshzeng, cslzhang}@comp.polyu.edu.hk

## Abstract

*Video super-resolution (VSR) aims to improve the spatial resolution of low-resolution (LR) videos. Existing VSR methods are mostly trained and evaluated on synthetic datasets, where the LR videos are uniformly downsampled from their high-resolution (HR) counterparts by some simple operators (e.g., bicubic downsampling). Such simple synthetic degradation models, however, cannot well describe the complex degradation processes in real-world videos, and thus the trained VSR models become ineffective in real-world applications. As an attempt to bridge the gap, we build a real-world video super-resolution (RealVSR) dataset by capturing paired LR-HR video sequences using the multi-camera system of iPhone 11 Pro Max. Since the LR-HR video pairs are captured by two separate cameras, there are inevitably certain misalignment and luminance/color differences between them. To more robustly train the VSR model and recover more details from the LR inputs, we convert the LR-HR videos into YCbCr space and decompose the luminance channel into a Laplacian pyramid, and then apply different loss functions to different components. Experiments validate that VSR models trained on our RealVSR dataset demonstrate better visual quality than those trained on synthetic datasets under real-world settings. They also exhibit good generalization capability in cross-camera tests. The dataset and code can be found at https://github.com/IanYeung/RealVSR.*

## 1. Introduction

Super-resolution (SR) [5] is a classical yet challenging task in image/video processing and computer vision, aiming at reconstructing high-resolution (HR) images/videos from their low-resolution (LR) counterparts. There are two major research branches in the field of SR: single image super-resolution (SISR) [9] and video super-resolution (VSR) [1]. While SISR mainly exploits the spatial redundancy within an image, VSR utilizes both spatial and temporal redundancies to reconstruct the HR video. With the increasing popularity of mobile imaging devices and rapid development of communication technology, VSR is attracting more and more attention for its great potentials in HR video generation and enhancement.



LR Vimeo-90k RealVSR+$\mathcal{L}_{v1}$ RealVSR+$\mathcal{L}_{v2}$

Figure 1. Video super-resolution results on a real-world video (captured by the iPhone 11 Pro Max) by EDVR [27] trained on the synthetic Vimeo-90k datatset [31] and our RealVSR dataset.

The recent progress in VSR research largely attributes to the rapid development of deep convolutional neural networks (CNNs) [3, 31, 14, 20, 25, 27, 11], which set new state-of-the-arts on several benchmarking VSR datasets [31, 23]. Those datasets, however, are mostly synthetic ones because it was difficult to collect real-world LR-HR video pairs. Specifically, the LR videos are obtained by uniformly downsampling their HR counterparts using some simple operators, e.g., bicubic downsampling or direct downsampling after Gaussian smoothing. Unfortunately, such simple degradation models could not faithfully describe the complex degradation processes in real-world LR videos. As a

[*] Equal contribution.

[†] Corresponding author. This work is supported by the Hong Kong RGC RIF grant (R5001-18).

result, VSR models trained on such synthetic datasets become much less effective when applied in real-world applications. An example is shown in Fig. 1, where we can see that the VSR model trained on the widely used Vimeo-90k datatset [31] is less effective in super-resolving on a real-world video captured by the iPhone 11 Pro Max.

In order to remedy the above mentioned problem, it is highly desired that we can have a VSR dataset of paired LR-HR sequences which are more consistent with the real-world degradations. Constructing such a paired dataset used to be very difficult since it requires capturing accurately aligned LR-HR sequences of the same dynamic scene simultaneously. Fortunately, the multi-camera system of iPhone 11 Pro series enables us to move one large step towards this goal. As shown in Fig. 2, there are three separate cameras of different focal lengths available in iPhone 11 Pro series. Utilizing the double taking function provided by the DoubleTake app, we are able to capture two approximately synchronized sequences using two of the three cameras. Some image registration algorithms [4] can then be employed to align the LR-HR video sequence pairs. Fig. 2 also shows an example of the LR-HR pairs before and after registration. In this way, a real-world VSR dataset, namely RealVSR, is constructed by capturing various indoor and outdoor scenes under different illuminations. RealVSR provides a worthy benchmark for training and evaluating VSR algorithms for real-world degradations.

Due to the constraints in dual camera capturing, there exists certain misalignment and luminance/color differences between the LR-HR sequences even after registration. Therefore, directly training a CNN to map the LR sequence to the HR sequence with simple losses is not a very suitable strategy. To alleviate the influence of color difference, we disentangle the luminance and color by transforming the RGB videos into YCbCr space, and focus on the reconstruction of video details such as edges and textures. On the color channels, we adopt a gradient weighted loss [30], intending to pay more attention to color edge reconstruction. To address the problem of small misalignment and luminance difference in Y channel, we decompose Y channels of predicted and targeted frames into Laplacian pyramids, and apply different losses on low-frequency and high-frequency components. As shown in Fig. 1, the VSR model trained on our dataset with the proposed learning strategy reproduces much better video details with less artifacts.

The contributions of this work are twofold. First, a RealVSR dataset (the first of its kind to the best of our knowledge) is constructed to mitigate the limitations of synthetic VSR datasets and provides a new benchmark for training and evaluating real-world VSR algorithms. Second, we propose a specific training strategy on RealVSR to learn VSR model with focus on detail reconstruction. Extensive experiments are conducted to validate the proposed RealVSR dataset and training strategy. Although the RealVSR dataset is built with iPhone 11 Pro Max, the VSR models trained on it also exhibit good generalization capability to videos captured by other mobile phone cameras.

## 2. Related Work

**Video super-resolution datasets.** There are several datasets widely adopted in the VSR research. Vimeo-90k [31] is the most popular one, which consists of more than 90,000 septuplets collecting from the Internet. Each septuplet contains 7 frames of resultion $256 \times 448$. REDS [23] is a dataset captured by the GOPRO sport camera. It consists of 300 sequences and each sequence contains 100 frames of resolution $720 \times 1280$. There are also some private datasets for VSR training [25]. In all these datasets, LR sequences are synthesized from HR sequences with simple degradation models, such as bicubic downsampling or direct downsampling after Gaussian smoothing. Though these datasets can serve as reasonable benchmarks for investigating and evaluating VSR algorithms, the adopted simple degradation model for LR-HR video pair generation makes them hard to use in practice because the degradation process of real-world videos is much more complex. When applying the VSR models trained on these datasets to real-world LR videos, the super-resolved videos are often over-smooth and prone to visual artifacts. This motivates us to build a real-world VSR dataset to narrow this synthetic-to-real gap.

**Real-world image super-resolution datasets.** Though there is no real-world VSR dataset publically available yet, several real-world SISR datasets have been built and released. Chen et al. [6] collected 100 LR-HR image pairs of printed postcards in a carefully controlled indoor environment. Zhang et al. [33] built a raw image SISR dataset of 500 outdoor scenes via optical zooming, while the LR-HR image pairs are not well-aligned. Cai et al. [4] constructed a real-world SISR benchmark also by optical zooming, but they developed a registration algorithm to carefully align the LR-HR image pairs so that end-to-end training of CNN is easy to implement. Wei et al. [30] further explored this idea and established a larger benchmarking dataset with more DSLR cameras. Inspired by those works on real-world SISR datasets, we propose to build the first real-world VSR dataset to facilitate the research of practical VSR.

**Video super-resolution methods.** The recent development of VSR algorithms [3, 31, 14, 20, 25, 27, 11] largely benefits from the rapid development of deep-learning technologies. Existing VSR algorithms can be roughly divided into two categories based on how the frame alignment is done. The first category of algorithms does not have an explicit alignment process. Instead, they resort to techniques such as 3-dimensional convolution [15] and recurrent neural network [11] to exploit spatial-temporal information. The other category of algorithms adopts explicit alignment to

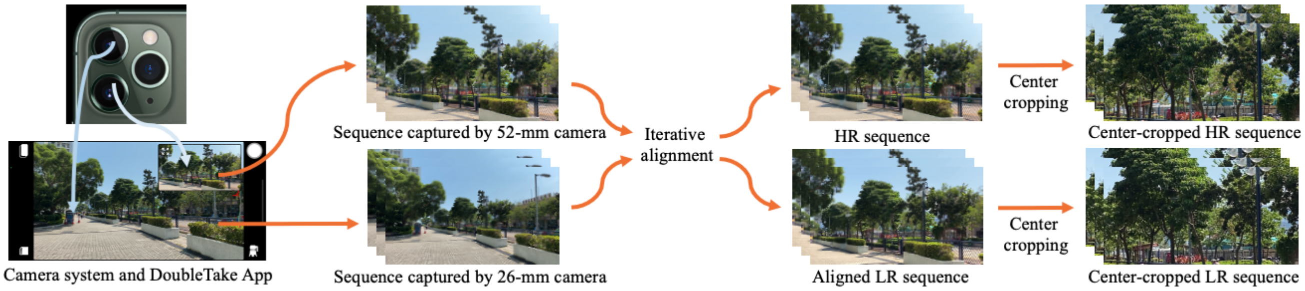

Camera system and DoubleTake App
Sequence captured by 52-mm camera
Sequence captured by 26-mm camera
Iterative alignment
HR sequence
Aligned LR sequence
Center cropping
Center cropping
Center-cropped HR sequence
Center-cropped LR sequence

Figure 2. The camera system of iPhone 11 Pro Max, the DoubleTake app, the captured low-resolution (LR) and high-resolution (HR) sequences, and the LR-HR video registration process.

help the network better exploit spatial-temporal information. These algorithms generally follow the paradigm of alignment, fusion and reconstruction. Earlier algorithms adopt optical flow to perform frame alignment. Deep-DE [20] and VSRnet [14] first perform motion compensation with optical flow and then reconstruct the HR frame with a CNN. Later, Caballero et al. [3] proposed an end-to-end solution called VESPCN, which integrates alignment and reconstruction into a single deep-learning framework. Similar strategies are adopted in DRVSR [25] and TOF [31]. Recently, deformable convolution [7, 35] has become popular for alignment owing to its powerful modeling capability. In particular, EDVR [27] aligns multi-level frame features with deformable convolution and fuses the aligned features with spatial and temporal attention. All the above VSR algorithms are developed based on the synthetic datasets. In this work, we built a real-world VSR dataset, which has distinct properties from the synthetic ones. Some new training strategies will be accordingly proposed to train effective real-world VSR models.

## 3. The Real-world VSR Dataset

Our goal is to build a real-world VSR dataset of paired LR-HR sequences, which can serve as a worthy benchmark to train and evaluate real-world VSR algorithms. The dataset is constructed using iPhone 11 Pro Max mobile phones with dual camera taking function provided by the DoubleTake app. As illustrated in Fig. 2, the DoubleTake app makes it possible to capture two approximately synchronized high-definite video sequences at different scales by two cameras with different focal lengths. There are three rear cameras mounted on iPhone 11 Pro Max: an ultra-wide camera with 13mm-equivalent lens, a wide camera with 26mm-equivalent lens, and a telephoto-camera with 52mm-equivalent lens. All the three cameras capture photos with 12 megapixels. Cameras with larger focal length can capture scenes with finer details, and the scaling factor is equal to the ratio of focal lengths. Considering the severe distortion of ultra-wide lens and the inferior image quality after cropping, we adopt the cameras with 26mm-equivalent lens and 52mm-equivalent lens for dataset construction. For each pair of captured video sequences, the sequence captured by camera with 52mm-equivalent lens is taken as the ground truth HR sequence, while the sequence captured by camera with 26mm-equivalent lens is adopted to generate the corresponding LR sequence, leading to a dataset for ×2 VSR. It is worth mentioning that ×2 scaling is currently the primary demand in practical VSR.

Using iPhone 11 Pro Max cameras and the DoubleTake app, we captured more than 700 sequence pairs. Each pair consists of two approximately synchronized sequences of frame rate 30fps and resolution 1080P. To ensure the diversity of the dataset, the captured sequences cover a variety of scenes, including outdoor and indoor scenes, daytime and nighttime scenes, still scenes and scenes with moving objects, etc. In general, scenes with rich textures are preferred as they are more effective to train a useful VSR model. The sequences in the dataset cover a variety of motions, including camera motions and object motions. After data collection, we manually selected and excluded about 200 sequences of inferior quality, e.g., severely blurred, noisy, over-exposed or under-exposed videos, etc. Considering the imperfect synchronization between the LR-HR sequences, we excluded sequence pairs with serious misalignment problem. After careful selection, 500 sequence pairs remain in the dataset. Fig. 3 shows some example scenes and the motion statistics of the dataset. More example scenes and content analysis can be found in the supplementary material.

Finally, the LR frames and HR frames in each sequence pair need to be aligned so that one can perform supervised VSR model training more easily. We adopt the image registration algorithm proposed in [4] to align the LR-HR videos frame by frame. Considering that there can be some small registration drifts between adjacent frames, we extended the registration algorithm in [4] by using five adjacent frames as inputs to compute the registration matrix of the centered frame. Once aligned, we crop the aligned LR and HR sequences at the center region of size $1024 \times 512$ to eliminate the alignment artifacts around the boundary. Fig. 2 illustrates the dataset construction process. It is worth noting

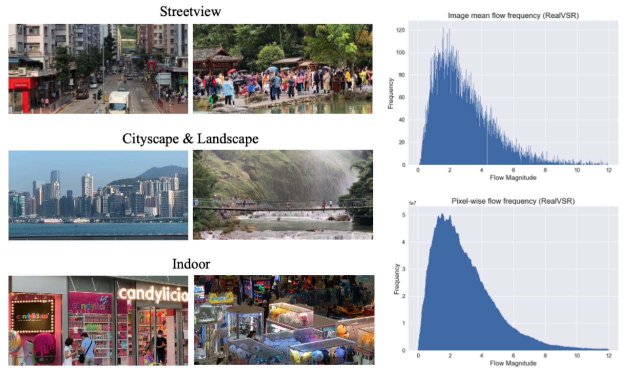


Figure 3. Example video scenes and motion statistics of the constructed RealVSR dataset.

that the LR and HR sequences are of the same size after registration. To further standardize the dataset, we cut all sequences to have the same length of 50 frames. The final dataset consists of 500 LR-HR sequence pairs, each of which has 50 frames in length and $1024 \times 512$ pixels in size.

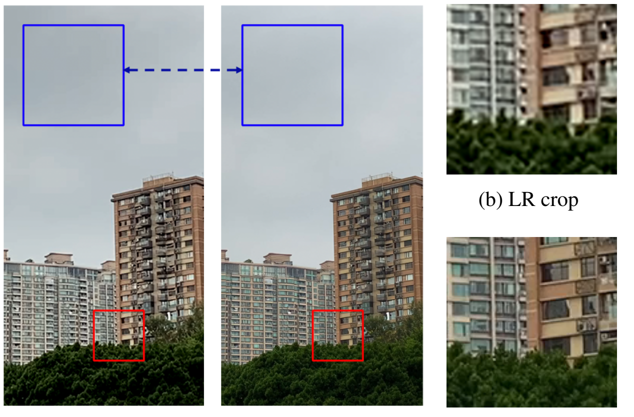
# 4. VSR Model Learning

## 4.1. Motivation and overall learning framework

Following most of the existing works [19, 31, 26, 27], we formulate VSR as a multi-frame super-resolution problem. Given $2N + 1$ consecutive LR video frames $\{I^L_{t-N}, ..., I^L_t ..., I^L_{t+N}\}$, we aim to predict the HR version of the center frame, denoted by $I^H_t$.

There are a few sources of image degradation in video acquisition process, such as the anisotropic blurring, the signal-dependent noise, the non-linear mapping in image signal processing (ISP) pipeline and the video compression algorithm, etc. Compared with the existing synthetic VSR datasets [31, 23] which assume the simple bicubic downsampling degradation, our RealVSR dataset is collected in the real-world environment and it naturally considers the complex degradation factors in real scenarios. However, it also poses greater challenges to effectively train VSR models. Specifically, the LR-HR videos taken from the two cameras undergo different lens, sensors and ISP pipelines, and thus exhibit different distortions. The registration algorithm [4] we adopt could alleviate the problems; however, there still exist minor misalignment and luminance/color differences between the LR-HR sequences. Fig. 4 shows an example, where we can see the slight global luminance and color difference between the LR and HR frames due to the variations in illumination, exposure time and camera ISP between the two cameras.

Our goal is to recover the image details (edge, texture, etc.) in the LR frames but not the global luminance and colors. Therefore, we propose a set of decomposition based losses to learn an effective VSR model from the constructed RealVSR dataset. The overall learning framework is shown in Fig. 5. Any existing VSR networks can be adopted in our framework with the proposed training losses. We convert the estimated and ground-truth HR videos into the YCbCr space to disentangle the luminance and color, and apply different loss functions on different components. On the Y channel, we design a Laplacian pyramid based loss to help the network better reconstruct the details under minor luminance difference. On color channels Cb and Cr, we adopt a gradient weighted content loss to focus on color edges. To further enhance the visual quality of the reconstructed HR video, we propose a multi-scale edge-based GAN loss to guide the texture generation. The details of the losses will be introduced in the next section.



(b) LR crop

(a) LR-HR pair

(c) HR crop

Figure 4. Slight luminance and color difference exist in some LR-HR sequence pairs.

## 4.2. Decomposition based losses

**Laplacian pyramid based loss on luminance channel.** The Y channel contains most of the texture information of a video frame and it is crucial to reconstruct image details in VSR. The commonly used losses in VSR research [19, 31, 26, 27], such as $L_1$ loss, $L_2$ loss and Charbonnier loss [17], are sensitive to global luminance differences, and hence the VSR models trained using such losses may be distracted from learning image structures and details. To tackle this problem, we decompose the Y channel into a Laplacian pyramid [2]. The low-frequency component captures the global luminance and general structure of the original image, and the high-frequency components contain the multi-scale details of the original image. By applying different losses on the low-frequency and high-frequency components, we are able to achieve better detail reconstruction while allowing certain difference in global luminance.

Denote the predicted HR luminance channel and the ground-truth HR luminance channel by $\hat{Y}$ and $Y$ respectively. As shown in Fig. 5, we decompose them into a three-layer Laplacian pyramid, denoted by $\{\hat{S}_0, \hat{S}_1, \hat{S}_2\}$ and $\{S_0, S_1, S_2\}$, respectively, where $\hat{S}_0$ and $S_0$ refer to the

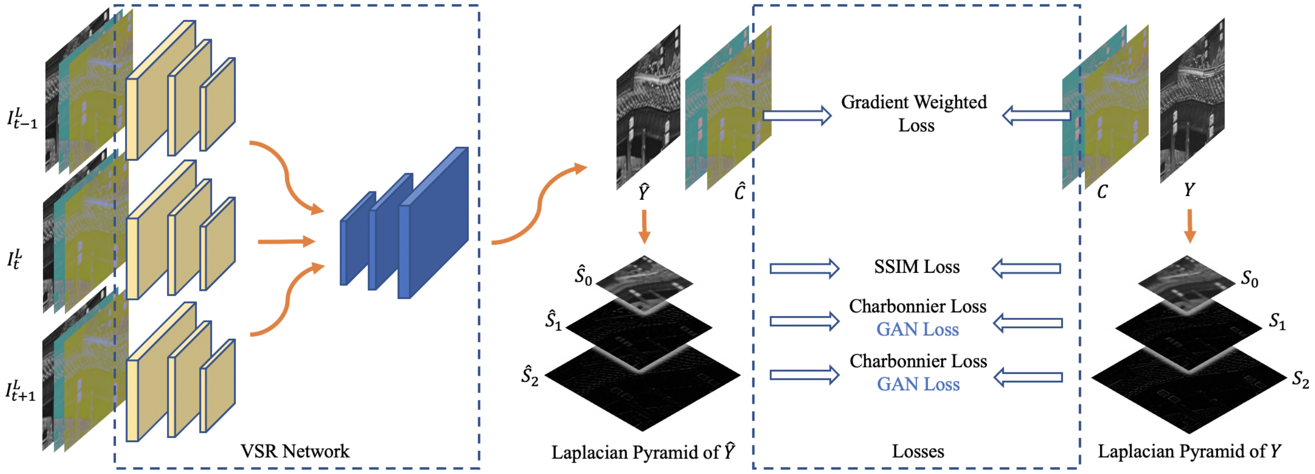


Figure 5. Framework of our decomposition based learning scheme of VSR models.

low-frequency components and others the high-frequency ones. The global luminance difference between $\hat{Y}$ and $Y$ lies mostly in the low-frequency components, and we adopt the SSIM loss [29] to encourage structural information reconstruction. Compared to the $L_1$ loss and $L_2$ loss, SSIM focuses more on the image structures and is insensitive to the luminance changes, which fits our goal well. The structure loss is given by

$$\mathcal{L}_s = \mathcal{L}_{\text{SSIM}}(\hat{S}_0, S_0) = 1 - \text{SSIM}(\hat{S}_0, S_0). \qquad (1)$$

Since the high-frequency components are basically free of global luminance difference, we adopt Charbonnier loss to encourage accurate reconstruction of fine details. The detail loss is then

$$\mathcal{L}_d = \sqrt{\left\|\hat{S}_2 - S_2\right\|^2 + \epsilon^2} + \sqrt{\left\|\hat{S}_1 - S_1\right\|^2 + \epsilon^2}, \qquad (2)$$

where $\epsilon = 10^{-3}$ is a small constant.

**Gradient weighted loss on chrominance channels.** Compared with the luminance channel, the chrominance channels CbCr are much smoother. We thus focus on reconstructing the prominent color edges on the chrominance channels. Inspired by [30], we adopt a gradient weighted loss here. Referring to Fig. 5, denote by $\hat{C}$ and $C$ the predicted and ground truth HR chrominance channels, respectively. The gradient weighted color loss is given by

$$\mathcal{L}_c = \sqrt{\left\|\Delta_{gw} * \hat{C} - \Delta_{gw} * C\right\|^2 + \epsilon^2}, \qquad (3)$$

where $\Delta_{gw} = (1 + w\Delta_x)(1 + w\Delta_y)$, $\Delta_x$ and $\Delta_y$ are the absolute difference maps between the gradient of $\hat{C}$ and $C$ in the horizontal and vertical directions, respectively. $w = 4$ is a weighting factor, and $\epsilon = 10^{-3}$ is a small constant.

**Multi-scale edge-based GAN loss.** The generative adversarial networks (GANs) [10] have been used in some SISR methods [18, 28] to improve the perceptual quality of estimated HR images. However, these methods usually apply GAN loss directly on the full-color image, which may not be effective enough to generate textures. We propose a multi-scale edge-based GAN loss by adopting the design of PatchGAN [12] and the relativistic average discriminator [13]. The GAN loss is applied to the high frequency components $S_1$ and $S_2$ of the Laplacian pyramid to enable better fine-grained discrimination for the VSR task. The adversarial loss for the generator is

$$\mathcal{L}_G = \sum_i \{-\mathbb{E}_{S_i}[\log(1 - D_i(S_i, \hat{S}_i))] - \mathbb{E}_{\hat{S}_i}[\log(D_i(\hat{S}_i, S_i))]\}, \qquad (4)$$

and the loss for the discriminator is

$$\mathcal{L}_D = \sum_i \{-\mathbb{E}_{S_i}[\log(D_i(S_i, \hat{S}_i))] - \mathbb{E}_{\hat{S}_i}[\log(1 - D_i(\hat{S}_i, S_i))]\}, \qquad (5)$$

where $D_i$ is the relativistic average discriminator for the $i$-th high-frequency components of the Laplacian pyramid.

**Final loss.** With the reconstruction losses $\mathcal{L}_s$, $\mathcal{L}_d$, $\mathcal{L}_c$ and adversarial loss $\mathcal{L}_{\text{adv}}$, we propose two versions of final loss for VSR network training. The first version, denoted by $\mathcal{L}_{\text{v1}}$, focuses on the reconstruction of fine details, which combines $\mathcal{L}_s$, $\mathcal{L}_d$ and $\mathcal{L}_c$ as follows:

$$\mathcal{L}_{\text{v1}} = \mathcal{L}_s + \mathcal{L}_d + \mathcal{L}_c, \qquad (6)$$

The second version, denoted by $\mathcal{L}_{\text{v2}}$, aims to further enhance the visual quality by generating some details, and is defined as

$$\mathcal{L}_{\text{v2}} = \mathcal{L}_{\text{v1}} + \lambda\mathcal{L}_{\text{adv}}, \qquad (7)$$

where $\mathcal{L}_{\text{adv}}$ is $\mathcal{L}_G$ for the generator and $\mathcal{L}_D$ for the discriminator, and $\lambda$ is a parameter to control to what degree the synthetic details will be involved.

# 5. Experiments

## 5.1. Experiment settings

**Datasets.** Apart from the constructed RealVSR, we also adopt the widely used synthetic Vimeo-90k [31] dataset in the experiments. Vimeo-90k consists of more than 90,000 7-frame sequences of resolution $256 \times 448$. Among them, 64,612 sequences are selected as the training set. The LR sequences in Vimeo-90k are synthesized via bicubic (BI) downsampling. Our RealVSR dataset consists of 500 real-world LR-HR sequence pairs with $1024 \times 512$ resolution. Each sequence contains 50 frames. We randomly select 50 sequence pairs as the testing set and leave the remaining 450 sequence pairs as the training set.

**VSR networks.** We conduct experiments by taking 5 representative and recently developed VSR models into our VSR model learning framework (referring to Fig. 5): RCAN [34], FSTRN [19], TOF [31], TDAN [26] and EDVR [27]. RCAN is a representative deep network for SISR. We modify it for VSR by concatenating the input frames along the channel dimension. FSTRN is a lightweight VSR model without explicit alignment. It exploits spatial-temporal information with separable 3D convolution. TOF is a typical VSR model which performs image domain alignment using optical flow. We replace its reconstruction branch with a residual backbone with 10 residual blocks. TDAN is a pioneer VSR model with deformable convolution [7]. EDVR is a powerful and popular VSR model which perform feature space alignment using deformable convolution. For EDVR, we adopt its moderate version and remove the TSA module, which mainly consists of a PCD alignment module and a reconstruction backbone with 10 residual blocks. For all methods, we remove their upsampling operations to fit our RealVSR dataset.

**Implementation details.** We randomly crop patches of size $192 \times 192$ from the video frames during training. The mini-batch size is set to 32. Data augmentation is performed by random horizontal flipping and random $90^{\circ}$ rotation. Moreover, we adopt the CutBlur [32] technique to alleviate the risk of overfitting in real-world VSR training. For the weighting factors in $\mathcal{L}_{\mathrm{v2}}$, we empirically set $\lambda = 1e^{-4}$. We choose Adam [16] as the optimizer with default parameters. For model training with $\mathcal{L}_{\mathrm{v1}}$, we set the initial learning rate to $1e^{-4}$. For model training with $\mathcal{L}_{\mathrm{v2}}$, we initialize the model weights with those trained with $\mathcal{L}_{\mathrm{v1}}$ and set the initial learning rate to $5e^{-5}$. In both cases, we gradually decay the learning rate with the cosine learning rate decay strategy. All the models are trained for $150,000$ iterations. We conduct all experiments with the PyTorch [24] framework.

## 5.2. Synthetic dataset vs. RealVSR dataset

To demonstrate the advantages of our dataset in real-world VSR, we compare the performance of VSR models trained on the synthetic Vimeo-90k dataset and our RealVSR dataset. With the 5 VSR networks (RCAN, FSTRN, TOF, TDAN, EDVR), 10 VSR models are trained in total on the two datasets. To balance the speed and performance, 3 adjacent LR frames are used to estimate the center HR frame. For fair comparison, we train all the 10 models with the baseline Charbonnier (CB) loss in YCbCr space.

We evaluate the 10 trained models on the RealVSR testing set. Table 1 lists the quantitative results in both full-reference and no-reference metrics. Considering the influence of slight color difference between LR and HR sequences in RealVSR, we compute the PSNR/SSIM indices on the Y channel to more accurately reflect the performance of texture reconstruction. As shown in Table 1, compared with the baseline bicubic interpolator (LR), VSR models trained on the synthetic dataset only achieve small improvement in terms of SSIM, while perform even worse in terms of PSNR. This validates that VSR models trained on synthetic dataset cannot generalize well to the real-world videos with more complex degradations. In contrast, all VSR models trained on our RealVSR dataset achieve much better performance in terms of PSNR/SSIM. We also compare the results with two popular no-reference image quality metrics, NIQE [22] and BRISQUE [21]. Models trained on RealVSR dataset also demonstrate better performance.

Fig. 6 shows the super-resolved frames for qualitative comparison. One can see that models trained on synthetic dataset tend to generate blurry edges and some artifacts, while models trained on RealVSR produce sharper edges and exhibit much less artifacts. This further demonstrates the importance of using data with real-world degradations for training a robust VSR model. More visual examples can be found in the supplementary file.

We further compare the temporal consistency of VSR results trained with different datasets. Models trained on the RealVSR dataset achieves better temporal consistency measured by vector norm differences of warped frames (T-diff).

## 5.3. Study on losses

In this section, we conduct experiments to demonstrate the effectiveness of the proposed losses $\mathcal{L}_{\mathrm{v1}}$ and $\mathcal{L}_{\mathrm{v2}}$. We use two representative VSR networks, TOF and EDVR, in this study. We train the models with 5 different losses. Three of them are fidelity-oriented losses. The first one is the baseline CB loss on YCbCr channels ($\mathcal{L}_{\mathrm{CB}}^{\mathrm{YCbCr}}$). The second one combines the proposed $\mathcal{L}_s$ and $\mathcal{L}_d$ on Y channel with the CB loss on CbCr channels ($\mathcal{L}_s$+$\mathcal{L}_d$+$\mathcal{L}_{\mathrm{CB}}^{\mathrm{CbCr}}$). The third one is our $\mathcal{L}_{\mathrm{v1}}$. The other two are perceptual-oriented losses. One is to combine $\mathcal{L}_{\mathrm{v1}}$ with the baseline RaGAN discriminator [28] on Y channel, denoted by $\mathcal{L}_{\mathrm{v1}}+\mathrm{RaGAN}$, and another one is our $\mathcal{L}_{\mathrm{v2}}$.

The TOF and EDVR models trained with the five different losses are evaluated on the RealVSR testing set, and

Table 1. Quantitative results of different VSR models evaluated on our RealVSR testing set.

| Metric | Bicubic (LR) | RCAN [34] | | FSTRN [19] | | TOF [31] | | TDAN [26] | | EDVR [27] | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Vimeo-90k | RealVSR | Vimeo-90k | RealVSR | Vimeo-90k | RealVSR | Vimeo-90k | RealVSR | Vimeo-90k | RealVSR |
| PSNR ↑ | 24.67 | 24.66 | 25.50 | 24.63 | 25.30 | 24.58 | 25.59 | 24.64 | 25.62 | 24.58 | 25.60 |
| SSIM ↑ | 0.7798 | 0.7836 | 0.8056 | 0.7880 | 0.8004 | 0.7884 | 0.8081 | 0.7848 | 0.8061 | 0.7838 | 0.8102 |
| NIQE ↓ | 5.0627 | 4.7269 | 4.0450 | 4.7071 | 4.1097 | 4.4749 | 3.9730 | 4.7090 | 4.1009 | 4.6377 | 3.9082 |
| BRISQUE ↓ | 43.1071 | 40.3198 | 36.0936 | 39.7353 | 36.2593 | 38.6576 | 35.4883 | 39.3448 | 36.2331 | 39.4596 | 34.9699 |
| T-diff ↓ | 3.9145 | 4.2561 | 3.6938 | 4.3796 | 3.8844 | 4.6101 | 3.7706 | 4.4174 | 3.8695 | 4.4229 | 3.6860 |

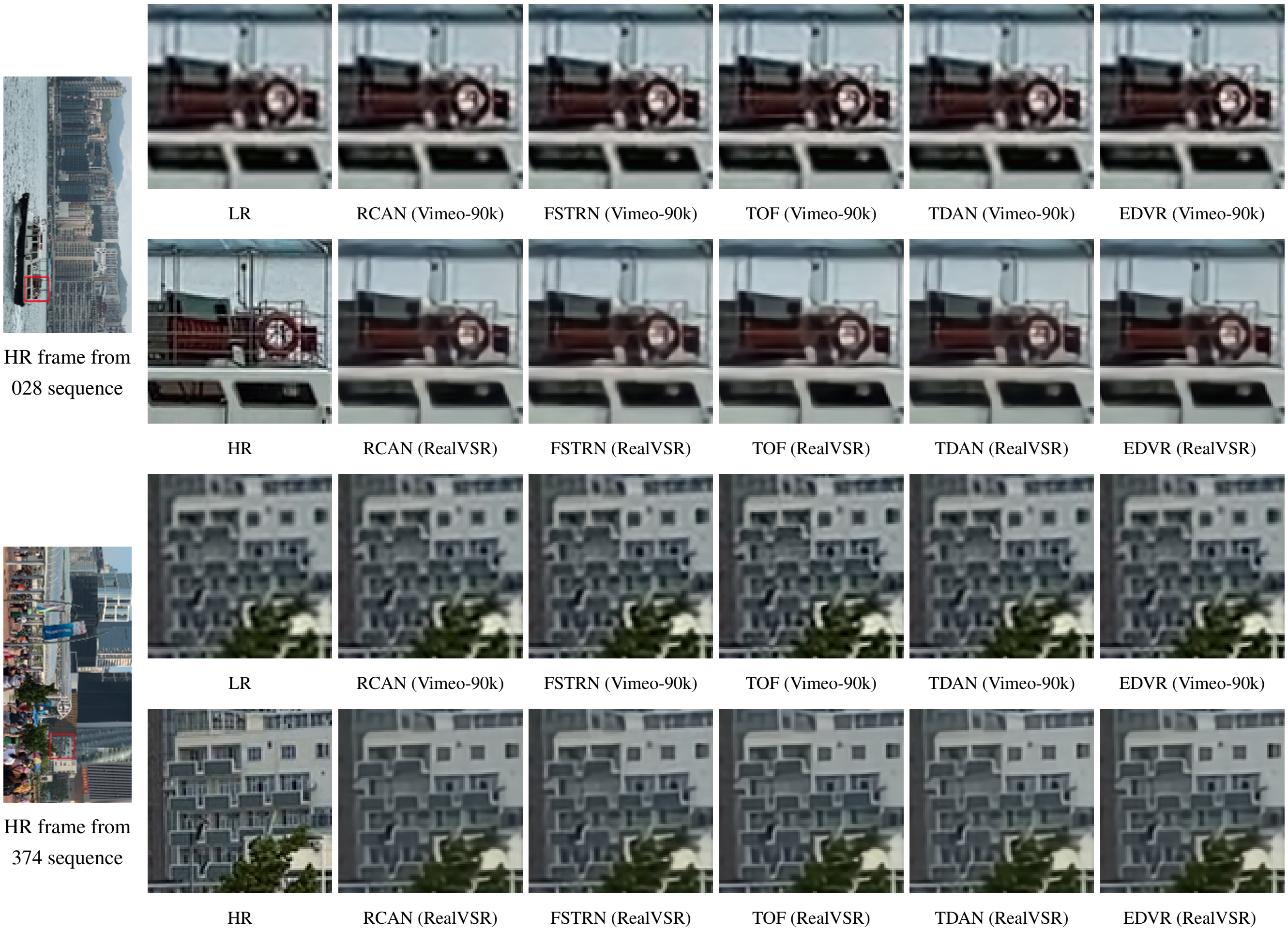
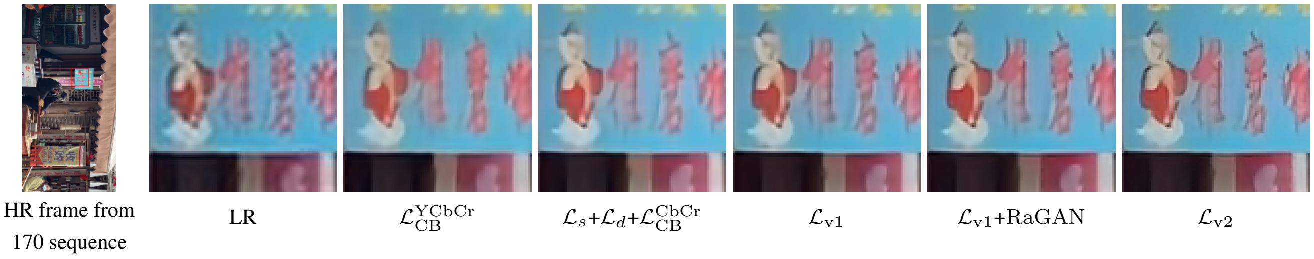


Figure 6. ×2 VSR results on our RealVSR testing set by different models.

the quantitative results are listed in Table 2. We evaluate the fidelity-oriented models by PSNR and SSIM and the perceptual-oriented models by LPIPS [34] and DISTS [8].

As shown in Table 2, models trained with $\mathcal{L}_s$+$\mathcal{L}_d$+$\mathcal{L}_{\mathrm{CB}}^{\mathrm{CbCr}}$ and $\mathcal{L}_{\mathrm{v1}}$ achieve better PSNR/SSIM results than those trained with baseline $\mathcal{L}_{\mathrm{CB}}^{\mathrm{YCbCr}}$. As we mentioned in Section 4.2, PSNR/SSIM may not be able to faithfully reflect the improvement of a VSR model considering the possible misalignment and luminance difference between the LR and HR sequences, while our losses in $\mathcal{L}_{\mathrm{v1}}$ aim to improve the frame details under these conditions but not only PSNR/SSIM. Therefore, we further visualize the VSR results obtained by the EDVR models in Fig. 7. One can see that, compared to the baseline, the proposed decomposition based losses ($\mathcal{L}_s$+$\mathcal{L}_d$+$\mathcal{L}_{\mathrm{CB}}^{\mathrm{CbCr}}$ and $\mathcal{L}_{\mathrm{v1}}$) help networks reconstruct sharper edges and more fine-scale details, showing better visual quality.

Regarding the perceptual-oriented models, referring to Table 2, our proposed $\mathcal{L}_{\mathrm{v2}}$ results in better LPIPS/DISTS scores than $\mathcal{L}_{\mathrm{v1}}$+RaGAN, demonstrating the role of multi-scale edge based discriminator. Regarding the qualitative comparison, as shown in Fig. 7, the proposed $\mathcal{L}_{\mathrm{v2}}$ enable networks to generate sharper details than $\mathcal{L}_{\mathrm{v1}}$+RaGAN. It also improves the visual quality of the results obtained by VSR models trained with $\mathcal{L}_{\mathrm{v1}}$. More visual examples can be found in the supplementary file.


HR frame from 170 sequence

LR

$\mathcal{L}_{\mathrm{CB}}^{\mathrm{YCbCr}}$

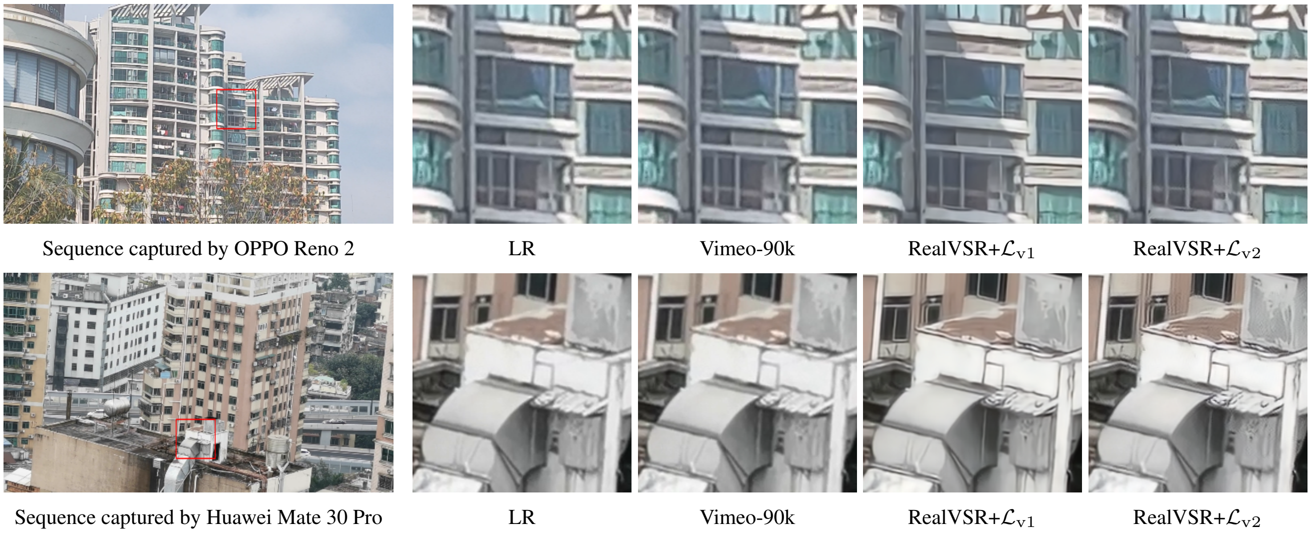
$\mathcal{L}_s$+$\mathcal{L}_d$+$\mathcal{L}_{\mathrm{CB}}^{\mathrm{CbCr}}$

$\mathcal{L}_{\mathrm{v}1}$

$\mathcal{L}_{\mathrm{v}1}$+RaGAN

$\mathcal{L}_{\mathrm{v}2}$

Figure 7. ×2 VSR results on videos from the RealVSR testing set by the EDVR [27] model trained with different losses.


Sequence captured by OPPO Reno 2

LR

Vimeo-90k

RealVSR+$\mathcal{L}_{\mathrm{v}1}$

RealVSR+$\mathcal{L}_{\mathrm{v}2}$

Sequence captured by Huawei Mate 30 Pro

LR

Vimeo-90k

RealVSR+$\mathcal{L}_{\mathrm{v}1}$

RealVSR+$\mathcal{L}_{\mathrm{v}2}$

Figure 8. ×2 VSR results on real-world videos outside RealVSR dataset by the EDVR [27] models trained on synthetic Vimeo-90k [31] and our RealVSR.

Table 2. Ablation studies on losses. PSNR/SSIM are evaluted on Y channel. LPIPS/DISTS are evaluated on RGB channels.

| Fidelity-oriented Comparison | | | | | | |
|---|---|---|---|---|---|---|
| Model | $\mathcal{L}_{\mathrm{CB}}^{\mathrm{YCbCr}}$ | | $\mathcal{L}_s$+$\mathcal{L}_d$+$\mathcal{L}_{\mathrm{CB}}^{\mathrm{CbCr}}$ | | $\mathcal{L}_{\mathrm{v}1}$ | |
| | PSNR ↑ | SSIM ↑ | PSNR ↑ | SSIM ↑ | PSNR ↑ | SSIM ↑ |
| TOF [31] | 25.59 | 0.8081 | 25.65 | 0.8110 | 25.66 | 0.8115 |
| EDVR [27] | 25.60 | 0.8102 | 25.83 | 0.8130 | 25.83 | 0.8131 |

| Perception-oriented Comparison | | | | | | |
|---|---|---|---|---|---|---|
| Model | $\mathcal{L}_{\mathrm{v}1}$ | | $\mathcal{L}_{\mathrm{v}1}$+RaGAN | | $\mathcal{L}_{\mathrm{v}2}$ | |
| | LPIPS ↓ | DISTS ↓ | LPIPS ↓ | DISTS ↓ | LPIPS ↓ | DISTS ↓ |
| TOF [31] | 0.2636 | 0.0857 | 0.2625 | 0.0861 | 0.2622 | 0.0810 |
| EDVR [27] | 0.2612 | 0.0869 | 0.2598 | 0.0852 | 0.2459 | 0.0766 |

### 5.4. Real-world video testing

To further demonstrate the advantages of our RealVSR dataset and the proposed training losses, we evaluate the trained models on several real-world videos outside the dataset. The testing videos are captured by several models of mobile phone cameras. The VSR results by the EDVR [27] models trained on Vimeo-90k [31] and RealVSR are shown in Fig. 1 and Fig. 8. Compared with the model trained on the synthetic Vimeo-90k dataset, the model trained on our RealVSR dataset with loss $\mathcal{L}_{\mathrm{v}1}$ reconstructs clearer edges with less artifacts. In addition, the model trained with loss $\mathcal{L}_{\mathrm{v}2}$ enriches the details and textures, further improving the visual quality. More visual examples and video demonstrations can be found in the supplementary file.

## 6. Conclusion

In this paper, we built the first, to our best knowledge, real-world VSR dataset with paired LR-HR sequences of various scenes, attempting to bridge the synthetic-to-real gap in VSR research and provide a benchmark for training and evaluating different VSR algorithms. Considering the inevitable minor misalignment and luminance/color difference between the captured LR-HR sequences, we proposed a Laplacian pyramid based loss to help the VSR networks better reconstruct video frame details. We further proposed a multi-scale edge based discriminator to guide the detail and texture generation and enhance the visual quality of the generated HR sequences. Our experiments demonstrated that VSR models trained on our dataset with the proposed learning scheme exhibit better visual quality on real-world videos than those trained on synthetic datasets. They can also be generalized to videos captured by other mobile phone cameras.

## References

[1] Christopher M Bishop, Andrew Blake, and Bhaskara Marthi. Super-resolution enhancement of video. In *AISTATS*, 2003. 1

[2] Peter Burt and Edward Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on communications*, 31(4):532–540, 1983. 4

[3] 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. 1, 2, 3

[4] Jianrui Cai, Hui Zeng, Hongwei Yong, Zisheng Cao, and Lei Zhang. Toward real-world single image super-resolution: A new benchmark and a new model. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3086–3095, 2019. 2, 3, 4

[5] Subhasis Chaudhuri. *Super-resolution imaging*, volume 632. Springer Science & Business Media, 2001. 1

[6] Chang Chen, Zhiwei Xiong, Xinmei Tian, Zheng-Jun Zha, and Feng Wu. Camera lens super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1652–1660, 2019. 2

[7] 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, 6

[8] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P Simoncelli. Image quality assessment: Unifying structure and texture similarity. *arXiv preprint arXiv:2004.07728*, 2020. 7

[9] Daniel Glasner, Shai Bagon, and Michal Irani. Super-resolution from a single image. In *2009 IEEE 12th international conference on computer vision*, pages 349–356. IEEE, 2009. 1

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 5

[11] Yan Huang, Wei Wang, and Liang Wang. Video super-resolution via bidirectional recurrent convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):1015–1028, 2017. 1, 2

[12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 5

[13] Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan. *arXiv preprint arXiv:1807.00734*, 2018. 5

[14] Armin Kappeler, Seunghwan Yoo, Qiqin Dai, and Aggelos K Katsaggelos. Video super-resolution with convolutional neural networks. *IEEE Transactions on Computational Imaging*, 2(2):109–122, 2016. 1, 2, 3

[15] Soo Ye Kim, Jeongyeon Lim, Taeyoung Na, and Munchurl Kim. 3dsrnet: Video super-resolution using 3d convolutional neural networks. *arXiv preprint arXiv:1812.09079*, 2018. 2

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[17] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 624–632, 2017. 4

[18] 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. 5

[19] Sheng Li, Fengxiang He, Bo Du, Lefei Zhang, Yonghao Xu, and Dacheng Tao. Fast spatio-temporal residual network for video super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10522–10531, 2019. 4, 6, 7

[20] Renjie Liao, Xin Tao, Ruiyu Li, Ziyang Ma, and Jiaya Jia. Video super-resolution via deep draft-ensemble learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 531–539, 2015. 1, 2, 3

[21] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on image processing*, 21(12):4695–4708, 2012. 6

[22] Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal processing letters*, 20(3):209–212, 2012. 6

[23] 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 Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 1, 2, 4

[24] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017. 6

[25] 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. 1, 2, 3

[26] 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. 4, 6, 7

[27] 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 Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 1, 2, 3, 4, 6, 7, 8

[28] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018. 5, 6

[29] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 5

[30] 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. *arXiv preprint arXiv:2008.01928*, 2020. 2, 5

[31] 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. 1, 2, 3, 4, 6, 7, 8

[32] Jaejun Yoo, Namhyuk Ahn, and Kyung-Ah Sohn. Rethinking data augmentation for image super-resolution: A comprehensive analysis and a new strategy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8375–8384, 2020. 6

[33] Xuaner Zhang, Qifeng Chen, Ren Ng, and Vladlen Koltun. Zoom to learn, learn to zoom. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3762–3770, 2019. 2

[34] 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. 6, 7

[35] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9308–9316, 2019. 3