RealPixVSR: Pixel-Level Visual Representation Informed Super-Resolution of Real-World Videos

Tony Nokap Park
SK Telecom
Seoul,Korea
tony.nokap.park@sk.com

Yunho Jeon
Hanbat National University
Daejeon,Korea
yhjeon@hanbat.ac.kr

Taeyoung Na
SK Telecom
Seoul,Korea
taeyong.na@sk.com

Abstract

Recently, there have been significant advances in video super-resolution (VSR) techniques under blind and practical degradation settings. These techniques restore the fine details of each video frame while maintaining the temporal consistency between frames for a smooth motion. Unfortunately, many attempts still fall short in the case of real-world videos. When diverse and complex in-the-wild degradation is introduced, the task becomes non-trivial and challenging. As a result, VSR techniques perform poorly in general. We argue that there is more space to improve the performance of VSR methods, as current methods are only trained on image-level degradation settings, leading to a restoration quality that may be sub-optimal for real-world degradation that varies pixel-wise within an image. To this end, we propose RealPixVSR which leverages the pixel-level representations to improve the pixel-level sensitivity to degradation. The pixel-level content-invariant degradation representation is learned in a self-supervised manner using the contrastive learning network referred to as the Pixel-Degradation-Representation-Network (PDRN). And the learned visual representation is merged with the cleaning and restoration networks using the Pixel-Degradation-Informed-Block (PDIB). Through experiments, we show that our network outperforms the latest state-of-the-art VSR models for real-world video.

1. Introduction

Video super-resolution(VSR) is the task of reconstructing high-resolution videos from low-quality videos containing degradations. It leverages the long-term information from neighboring frames to restore each frame in a given video. Two frameworks are mostly employed for the aggregation of frame information. Sliding-windows framework [14, 36, 41] is a branch of approach that uses the features in images within a short temporal window. Recurrent framework [3, 4, 33] is another approach that exploits long-term dependency where through recurrence the latent features are propagated and aggregated. Recently, [3] proposes a basic but powerful recurrent architecture that allows an easy extension.

The VSR task may become very challenging depending on the degradation in videos. If the goal is to deal with all real-world videos containing a complex combination of degradations, designing a dataset for the training itself becomes almost prohibitive as we must collect LR videos containing all combinations of different degradations and their corresponding HR videos. Hence, early classical degradation models [1, 12, 26, 28] assume a simplified degradation process to super-resolve videos under mild degradations. Although they achieved some success, the performance drop is significant with real images. Later, practical models [40, 53] designed a more complex high-order degradation process to increase their performance on real images but still, some performance gap exists. In [4], the authors extend [3] with a cleaning module and achieved some improvement over practical models in super-resolving real-world videos.

Contrastive learning is a method for unsupervised learning of visual representation which has achieved remarkable success in transfer performance recently. Its objective is to learn hidden representations by contrasting positive pairs with negative ones. Some research efforts [24, 43, 46, 51] have been conducted to apply contrastive learning to single image super-resolution (SISR). For instance, [24] applied contrastive learning to learn degradation representation to distinguish the latent degradation from one another without any explicit degradation assumption. The learned degradation representation is then passed to a degradation-aware network(DASR) with flexible adaptation to various degradations to super-resolve an HR image.

Although these methods show the effectiveness of contrastive learning on SISR tasks, they impose a strong assumption that the degradation representation is fixed at the image level. This assumption may lead to a restoration qual-

ity that is sub-optimal for processing real-world degradation that also varies pixel-wise within an image. When the degradation representation is fixed at the image level, the discrepancy between image and pixel level representation induces some quality loss in the image reconstruction. For VSR tasks, the representation error propagates during the temporal aggregation of features and may result in an accumulated quality loss in the reconstructed video.

In this paper, we propose a recurrent method for video super-resolution of real videos that leverages the degradation representation learned through unsupervised pixel-level contrastive learning. We relax the assumption of image-level representation and extend the degradation representation to pixel-level to enforce the spatial sensitivity that we think benefits the reconstruction process. The overall architecture of our work (RealPixVSR) is depicted in Fig. 1. It is composed of three main components: the pixel degradation representation network (PDRN), the pixel degradation informed cleaning network (PDICN) and the pixel degradation informed VSR network (PDIVSR). The contributions of our work can be summarized as follows:

- We propose a novel recurrent structure for the propagation of pixel-level degradation representation. Our model predicts the degradation representation at the pixel level and demonstrates improved restoration of real images
- Compared to previous state-of-the-art methods, our experiments show that our method achieves better performance on real-world videos while preserving image details.

2. Related Work

Single Image Super-Resolution Given a low resolution(LR) image x, single-image super-resolution (SISR) method predicts approximation image y' of a high-resolution(HR) image y, by

$$y' = f_{sr}(x, \theta) \tag{1}$$

where f_{sr} denotes the super-resolution model and θ is the corresponding parameter set. Generally, the LR image x is synthetically computed from HR image y by

$$x = f_{deg}(y, \tau) \tag{2}$$

where f_{deg} denotes a degradation process, and τ its parameters. f_{deg} may be composed of several degradation factors such as blurring, downsampling, noise injection, compression artifacts, anisotropic degradations, sensor noise, speckle noise, and more.

Non-blind, Blind, Practical vs Real-world SR Super-resolution can be classified into different categories based on the assumptions made about the degradation process.

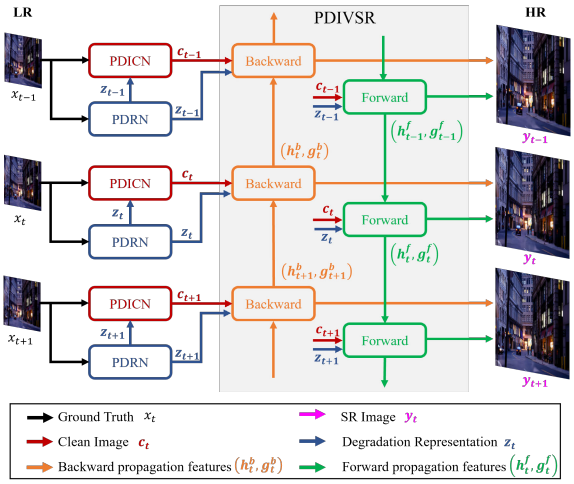


Figure 1. The overall architecture of RealPixVSR

Non-blind super-resolution models, such as [13, 15, 18], assume that f_{deg} and parameter τ in equation (2) are fixed. These models usually use a bicubic downsampling kernel as f_{deg} and do not consider other degradation factors. This assumption is too simple to reflect real-world degradation.
Blind super-resolution [12, 38, 40, 54] mimics more closely to real degradation by setting degradation parameters τ vary. For example, classical degradation models, define the whole degradation process as

$$x = (y \otimes k) \downarrow_s + n \tag{3}$$

where k denotes the blur kernel, $\otimes$ the convolution, $\downarrow_s$ the down-sampling operations(s is scaling factor) and n the noise restricted to simple cases. While more relaxed than non-blind models, this approach still handles only a limited range of degradation types.
Practical super-resolution models handle complex degradation processes, often expressed as a combination of multiple factors such as blur, sensor noise, sharpening artifacts, JPEG compression, etc. This requires high-order degradation models, like the one introduced in [52], which shuffles and applies multiple degradation types to HR images to generate a more realistic LR image.
Real-world super-resolution shifts the focus from synthetic to real images. [23] shows how the performance of SISR methods trained on synthetic data degrade due to the domain gap between synthetic and real-world data. Much research is being conducted on topics involving real-world dataset collection, SR models for real-world images, and SR result assessment.
Video Super-Resolution Video Super-Resolution (VSR) aims to generate high-resolution detailed video frames from a given set of low-resolution video frames. Frame-wise the task is similar to SISR, but VSR also leverages visual information available from a frame's neighbors. To use the

information from a neighbor frame, VSR methods exploit the temporal consistency with the neighbor frames during the reconstruction of HR frame to guarantee a smooth motion. Recurrent framework [3, 4, 33] is popularly used for feature aggregation in VSR models since it can process inputs of arbitrary length through weight sharing while handling long-range dependency among pixels. The sliding-windows framework [14, 36, 41] is also popularly used but it is known that for the sliding-windows method, the performance drops for smaller windows sizes. Either framework needs an alignment module to find the correspondence between pixels of an image and its neighbor to align neighbor frames through back-warping to the given image. The optical flow method, such as [32], is an efficient way to implement an alignment module. Deformable Alignment method [37, 41] is superior in performance and many recent works adopt it for feature alignment.

Contrastive Learning Contrastive learning [9, 17, 34, 47, 49, 55] is a method of unsupervised representation learning in a dataset without the need for labels. It works by maximizing the similarity between positive pairs and maximizing the difference between negative pairs, generated through data augmentation methods like image transformations [6, 16, 47], cropped sub-images [17, 39], or multiple views [35]. This approach mimics the way humans learn by grouping common attributes and discriminating unique ones. By reducing the need for labeling, contrastive learning can guide deep models to learn intrinsic representations in the data. Recent studies have shown that contrastive learning can outperform supervised methods on various downstream tasks, with MOCO [16] achieving superior transfer performance.

Some studies [43, 44, 46, 51] applied contrastive learning to image restoration and showed its effectiveness. DARS [40] developed an unsupervised degradation representation learning scheme for blind super-resolution (SR) without estimating the degradation process, by learning abstract representations of various image-level degradations. AirNet [24], accomplishes a different task of restoring multiple types of degradation using a contrastive network that knows the representation of different types of degradations (rainy, cloudy, ..etc) and guides the SR network to do the proper restoration regardless of the difference in degradation types.

Pixel-level contrastive learning in videos Previous research on image restoration [24, 40] assumed that degradation is consistent across the entire image, which may limit the algorithm's ability to handle real-world images with varying degradation at the pixel level. In video super-resolution, image-level and pixel-level differences can propagate during temporal aggregation of frame features. Pixel-level contrastive learning may help, but identifying positive pairs in videos with repetitive pixels in multiple frames can be challenging. Some prior works [19, 21, 25, 42] addressed this by using weak cycle consistency checks between forward and backward associations among pixels. In contrast, [48] provides a method to compute the ground-truth pixel-level correspondence directly from different views of a single image while gracefully combining pixel-level and instance-level representations. Our approach follows their work, computing both instance-level and pixel-level representations. For clarity, we use the term image-level representation instead of instance-level representation to distinguish it from pixel-level representation.

3. Proposed Method

In this section, we describe in detail our proposed method. The overall architecture is illustrated in Fig. 1. It is composed of three main components: the pixel degradation representation network $\mathcal{N}_{PDRN}$, the pixel degradation informed cleaning network $\mathcal{N}_{PDICN}$ and the pixel degradation informed VSR network $\mathcal{N}_{PDIVSR}$. Given video frame sequences $x_t,\ _{t=1,\dots,N}$, the pixel degradation representation network $\mathcal{N}_{PDRN}$ computes the pixel-level latent representations

$$z_t = \mathcal{N}_{PDRN}(x_t), \quad _{t=1,\dots,N}. \tag{4}$$

Then, x_t and z_t are fed into the pixel degradation informed cleaning network $\mathcal{N}_{PDICN}$ to produce the clean image sequences

$$c_t = \mathcal{N}_{PDICN}(x_t, z_t), \quad _{t=1,\dots,N}. \tag{5}$$

Finally, c_t and z_t is fed into the pixel degradation informed VSR network $\mathcal{N}_{PDIVSR}$ to produce the super-resolved frame sequences

$$y'_t = \mathcal{N}_{PDIVSR}(c_t, z_t), \quad _{t=1,\dots,N}. \tag{6}$$

In our work, we adopt the recurrent structure similar to BasicVSR [3] with modified feature aggregation blocks.

3.1. Pixel Degradation Representation Network

The objective of PDRN is to generate a pixel-level degradation representation z_t in each frame. This representation is utilized to improve the quality of image reconstruction in later stages by providing detailed information about the degradation at the pixel level.

PDRN comprises a ResNet-based encoder and three projection heads, as illustrated in Fig. 2(b). The first projection head calculates the pixel-level degradation representation z_t, while the second and third projection heads are employed for pixel-level and image-level contrastive learning, respectively.

Learning pixel-level representation We use pixel-level contrastive learning make RealPixVSR better at noticing

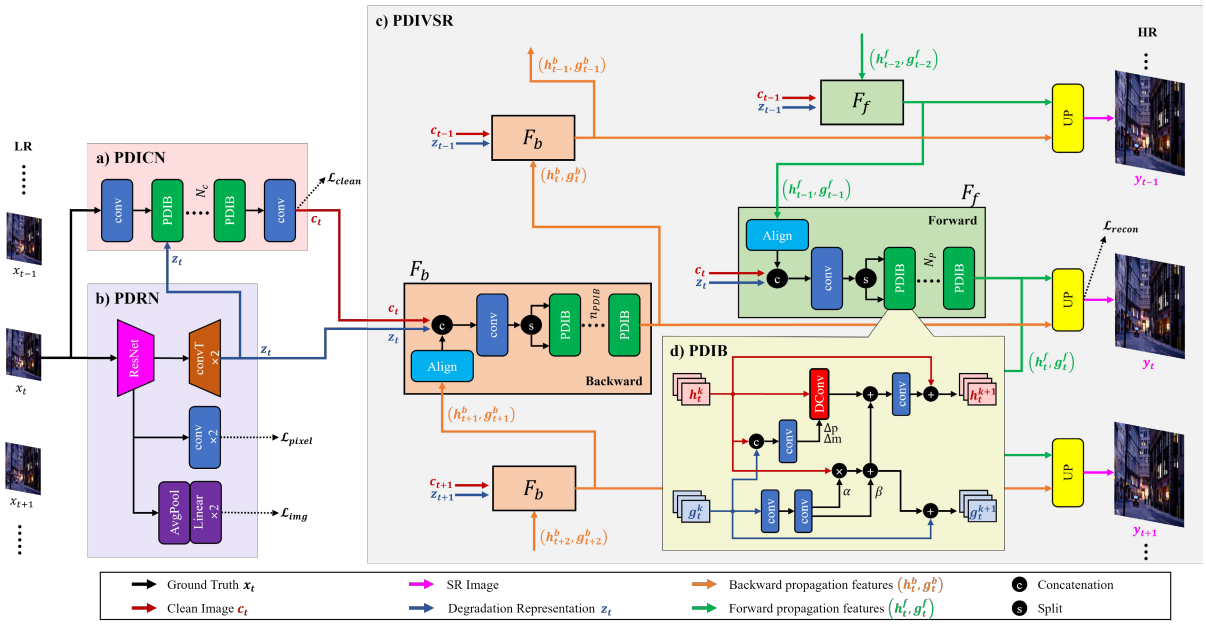


Figure 2. Model architecture of the proposed RealPixVSR. (a) Pixel Degradation informed cleaning network (PDICN) (b) Pixel Degradation Representation Network (PDRN) (c) Pixel Degradation Informed VSR Network (PDIVSR) (d) Pixel Degradation Informed Block (PDIB)

differences in pixel contents when there are various types of degradation present. This module selects two different views, V_1 and V_2, from a low-resolution (LR) image. This is achieved by randomly cropping patches from the image and resizing them back to their original size. One of these views is processed by an encoder network E and the other by a momentum encoder network $\tilde{E}$ to compute two different feature maps h^{V_1} and h^{V_2}, respectively. Similar to [48], we then calculate the normalized distance $dist(p, q)$ between pixel pairs $p \in V_1$ and $q \in V_2$ that takes into account the scale differences. We use a distance threshold, $\epsilon = 0.7$, to determine positive and negative pixel pairs, where pixels with a distance of less than or equal to ϵ are considered positive, and those with a distance greater than ϵ are considered negative.

Considering each pixel p as an instance, a contrastive loss $\mathcal{L}_{pixel}$ can be defined as follows:

$$\mathcal{L}_{pixel}(p) = -log \frac{\sum_{q \in \Psi_p^+} e^{\cos(h_p^{V_1}, h_q^{V_2})/\tau}}{\sum_{q \in \Psi_p^+} e^{\cos(h_p^{V_1}, h_q^{V_2})/\tau} + \sum_{q \in \Psi_p^-} e^{\cos(h_p^{V_1}, h_q^{V_2})/\tau}} \quad (7)$$

where $h_p^{V_1}$ and $h_q^{V_2}$ denote the feature vectors at pixel $p \in V_1$ and $q \in V_2$, respectively. The sets Ψ_p^+ and Ψ_p^- represent the positive and negative samples of pixel $p \in V_1$ in view V_2.

Image-level representation is also employed to complement pixel-level representation, as seen in Xie et al. [48]. Different projection heads are used to learn image-level and pixel-level representations separately, and features are aggregated using average pooling before projection is applied. The image-level contrastive loss $\mathcal{L}_{img}$ is computed using positive samples from the same frame and negative samples from different frames. The overall contrastive loss $\mathcal{L}_{cont}$ combines both pixel-level and image-level contrastive losses, as shown in Eq. (8), where α_{cont} represents a multiplicative factor that weights the image-level contrastive loss.

$$\mathcal{L}_{cont} = \mathcal{L}_{pixel} + \alpha_{cont}\mathcal{L}_{img}. \quad (8)$$

Contrastive learning primarily enhances the acquisition of detailed pixel-level representations, focusing less on representations related to degradation. Nevertheless, the representation z_t derived from PDRN is incorporated into PDICN and PDIVSR, both responsible for cleaning and reconstructing high-resolution (HR) images. This entire network is then trained end-to-end. Consequently, z_t is induced to capture the pixel-level degradation in images. Thus, the pixel-level contrast loss serves as a directive for PDRN to extract more intricate pixel-level representations of degradations. The effectiveness of this design is discussed in Sec. 5.2.3.

3.2. Pixel Degradation Informed Block

Pixel Degradation Informed Block (PDIB) is the primary component used to aggregate features in the cleaning and propagation layers. The structure of PDIB is shown in Fig. 2(d). The purpose of PDIB is utilizing pixel-level representation to make a detailed image features. To achieve

this, PDIB takes in two inputs: the image feature map h_t^k and the pixel-level representation g_t^k. In contrast to prior research [24], we also update the pixel-level representation to generate a fine-grained representation and propagate it to the next PDIB block. To ensure training stability, we update residual features, as depicted in Eq. (9)

$$\begin{aligned} h_t^{k+1} &= h_t^k + \mathcal{N}_H(h_t^k, g_t^k) \\ g_t^{k+1} &= g_t^k + \mathcal{N}_G(h_t^k, g_t^k), \end{aligned} \tag{9}$$

where $\mathcal{N}_H$ and $\mathcal{N}_G$ are the network for extracting image feature and pixel-level representation, respectively.

$\mathcal{N}_H$ is composed of a Deformable Convolution (DConv) layer [8] and a Spatial Feature Transform (SFT) layer [45] in order to effectively handle complex degradations. Deformable convolution has the ability to dynamically adjust the receptive field by adding 2D offsets Δp and a modulation mask Δm to the regular grid sampling locations in the standard convolution. The offsets are learned from the feature maps h_t^k and g_t^k via additional convolution layers. Real-world images often suffer from complex degradations that can vary from pixel to pixel. The DConv layer aims to find the optimal convolution settings within an image at the pixel level using pixel-level degradation information provided by g_t^k.

Both $\mathcal{N}_H$ and $\mathcal{N}_G$ utilize SFT, which can alter the distributions of h_t^k based on the value of g_t^k provided. This is achieved through convolution networks dedicated to computing and producing modulation parameters, α and β. This process effectively modifies the discrepancy in degradation distribution among individual pixels.

3.3. Pixel Degradation Informed Cleaning Network

The goal of PDICN is to reduce the impact of degradation on real images before they are processed by the VSR network, in order to minimize the influence of noise on downstream tasks. Previous research [4] has shown that if not handled properly, noise and artifacts can be amplified during the long-term aggregation of features. To address this issue, we propose an architecture that includes a pre-cleaning network that adopts pixel-level degradation representation, as shown in Fig. 2(a).

PDICN consists a stack of N_c PDIBs to output the cleaned image. Given an input image x_t, the first convolution layer extracts the initial image features h_t^0 for PDIB, and z_t from PDRN is used for the initial pixel representation for PDIB, i.e. $g_t^0 = z_t$. A last convolution layer generate the clean image c_t from the last image feature $h_t^{N_c}$ from PDIB. The loss for image cleaning $\mathcal{L}_{clean}$ is calculated from the difference between the output c_t and the bicubic-downsampled GT image $y_{t,\downarrow_4}$. We use Charbonnier loss [5] known to improve accuracy over L_2 loss.

3.4. Pixel Degradation Informed VSR Network

The PDIVSR is the main network where the super-resolution task is completed (Fig. 2(c)). It receives a clean input c_t from PDICN, and the pixel-level degradation representation z_t from PDRN. Given c_t and z_t, $_{t=1,\dots,N}$, PDIVSR processes clean LR image sequences to create super-resolved video frame sequences. We have used the framework proposed by [3] as a baseline and modified the structure to include the representations of pixel-level degradation. PDIVSR is composed of layers for backward and forward propagation, which propagate features in their respective directions. The upsampling layer located at the end of PDIVSR merges the output features from both propagation layers to generate the super-resolved image.

Propagation layers The proposed model implements a bi-directional propagation module for feature aggregation for a long-time leveraging of temporal information. The architecture of a propagation layer (forward or backward) consists of an initial convolution layer followed by a stack of PDIBs as shown in Fig. 2(c). For a backward propagation layer at time t, the inputs comprise of four components: the clean image c_t, the pixel-level degradation representation z_t, the recurrent feature map h_{t+1}^b, and the recurrent pixel-degradation feature map g_{t+1}^b. The feature maps h_{t+1}^b and g_{t+1}^b pass through the alignment module to line up the feature map at time $t+1$ to input c_t and z_t at the current time. Once aligned, all four elements are concatenated and fed into the initial convolution layer. The output feature map is then split into two feature maps h_t^0 and g_t^0, and they pass through N_p PDIB blocks to compute the feature maps $h_t^b = h_t^{N_p}$ and $g_t^b = g_t^{N_p}$. In summary, the feature propagation in both directions can be formulated as

$$\begin{aligned} h_t^b, g_t^b &= F_b(c_t, h_{t+1}^b, z_t, g_{t+1}^b), \\ h_t^f, g_t^f &= F_f(c_t, h_{t-1}^f, z_t, g_{t-1}^f), \end{aligned} \tag{10}$$

where F_b and F_f denote the backward and forward propagation layers, respectively.

Loss Function The entire network is trained end-to-end with two steps. For the first training step, we define the loss function as follows:

$$\mathcal{L}_{step1} = \mathcal{L}_{recon} + \mathcal{L}_{clean} + \mathcal{L}_{cont}. \tag{11}$$

The reconstruction loss $\mathcal{L}_{recon}$ is computed as the difference between the ground truth y_t and the reconstructed image y_t'. Charbonnier loss [5] is also used for the reconstruction loss, like $\mathcal{L}_{clean}$.

For the second training step, we fine-tune the network by adding the perceptual loss $\mathcal{L}_{gen}$ [20] and adversarial loss $\mathcal{L}_{disc}$ [10]:

$$\mathcal{L}_{step2} = \mathcal{L}_{step1} + \lambda_{gen}\mathcal{L}_{gen} + \lambda_{disc}\mathcal{L}_{disc}. \tag{12}$$

4. Training Details

In this section, we describe the dataset used in training, the loss functions, and the details of the training.

Dataset For training, we use REDS [31] as the HR dataset. The LR degradation dataset is generated by a two-step process. First, the method from Real-ESRGAN [44] is applied, and then the video compression follows [4]. Specifically, for the first step, we adopt the *second-order degradation process* [44]. We compute two iterations of a degradation process for each given frame. The degradation process at each iteration consists of a series of the random blur, resize, noise, and JPEG compression, with settings following [44]. Random blur is applied either isotropically or anisotropically using gaussian, generalized gaussian, or plateau bivariate kernels. For resize operations, the area, bilinear, or bicubic interpolation is applied. For random noise generation, the gaussian or Poisson kernel is used.

In contrast to single-image SR, the temporal dependency between frames must also be considered. As most videos on the internet are stored and streamed with compressed images at varying quality levels, video compression is applied at the last stage of the degradation sequence to append the spatio-temporal degradations related to compression algorithms. Following [4], we use FFmpeg with codecs selected at random among *h264*, *libx264*, and *mpeg4*, to compress the JPEG-compressed frames with the bit rate coefficient selected uniformly in between 10^4 and 10^5.

Training Details The training patch size of the HR image is set to 256×256 which is degraded into an LR image of size 64×64. We load 15 frames in each iteration and by using the stochastic degradation scheme [4] we double the frames in each batch to 30 temporal frame sequences to reduce the CPU bottleneck in loading images and increasing the training time while maintaining comparable accuracy.

For the data augmentation used in the PDRN, following the strategy in [11], two 32×32 patches are randomly cropped from each frame and resized to 64×64 followed by random horizontal flip, color distortion, Gaussian blur, and solarization.

The training is a two-step process. First, RealPixVSR is trained for 230K iterations. A total of 16 batches with each batch containing 30 temporal frames is used. Adam optimizer [22] is used with a learning rate fixed to 10^{-4}. To warm up, only the PDRN is trained first for 100 iterations using the contrastive loss $\mathcal{L}_{cont}$. Next, we finetune the network for 80K iterations using the weights from the first step. The reduced batch size of 8 is used. The learning rates are set to 2.5×10^{-5} and 10^{-4} for generator and discriminator, respectively. The scale parameters are set to $\lambda_{gen} = 1$ and $\lambda_{disc} = 5 \times 10^{-2}$.

Regarding the contrastive loss, the mixing coefficient α_{cont} is set to 0.1, and the scalar temperature hyperparameter τ is set to 0.3 following previous work [48]. The number of PDIB blocks for PDICN (N_c) and PDIVSR (N_p) are set to 19 and 18, respectively. All experiments were conducted in PyTorch on NVIDIA A100 GPUs.

5. Experiments

In this section, we describe the dataset used for experiments, the evaluation metrics, and the experiment results.

5.1. Datasets and Metrics

Experiment Dataset We use VideoLQ dataset [4] to assess the performance of RealPixVSR. Real-world dataset is hard to design and there is still a short amount of real-world datasets where many of them consist of a small amount of LR-HR pairs. VideoLQ consists of only real LR videos collected from various video-hosting sites with a Creative Common license. They selected videos of different resolutions and diverse contents to cover different types of degradation. Each video focuses only on one scene. The VideoLQ contains 50 short videos of 100 frames each with the exception of videos 30 to 33 containing less than 100 frames. We use UDM10 [50] for the validation test of our model. UDM consists of 10 short clips containing 32 frames each clip. It contains diverse scenes and relatively high-resolution frames.

Experiment Metrics Since the ground-truth HR videos for the VideoLQ dataset are not available, PSNR and SSIM cannot be computed for the VideoLQ dataset. Hence, we adopt the non-reference image quality assessment (IQA) metrics including NIQE [30], BRISQUE [29], NRQM [27] and PI [2] for quantitative evaluation.

5.2. Experiment Results

5.2.1 Quantitative Comparison

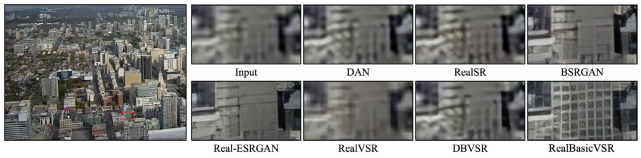
Tab. 1 presents the results of a quantitative comparison using no-reference IQA metrics on the VideoLQ dataset. Our proposed model, RealPixVSR, achieve the state-of-the-art performance on these metrics except for NIQE. All results for other models are taken from [4]. We evaluate each video by extracting its first, middle, and last frames and computed the IQA metrics on the Y-channel. We also reproduced the result of RealBasicVSR [4] using their official code [7] and checkpoint, but we got a slightly different result from theirs; in our experiment, RealPixVSR gave a better result even on NIQE.

Tab. 2 shows the results of the no-reference IQA metric on the UDM10 [50] dataset. Our method outperforms RealBasicVSR in all metrics, and the standard deviation of IQA values among frames is much smaller, indicating a consistent quality of the reconstructed frames. This suggests that our proposed recurrence structure is effective in accurately propagating features and representation maps. Comparing the computational complexity of the model, our model

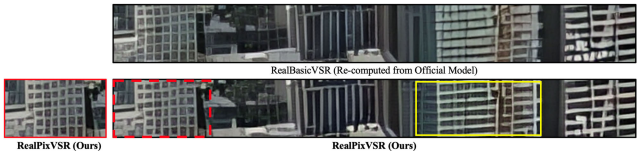
Table 1. **Quantitavie evaluation on VideoLQ [4]** A quantitative comparison using no-reference image quality assessment(IQA) metrics is shown. RealPixVSR achieves the best performance except for NIQE. Red and blue denote the best and second, respectively. Results for other models are from [4]. All metrics were computed on the Y-channel. For each video, three images were selected (first, middle, and last)

IQA	Bicubic	RealVSR	DAN	DBVSR	BSRGAN	R-ESRGSN	RealSR	RealBasicVSR*	RealPixVSR
NRQM [27]↑	2.8016	2.4958	3.3346	3.4097	5.7172	5.7108	5.6187	6.0477(6.0233)	6.1517
NIQE [30]]↓	8.0049	8.0606	7.1230	6.7866	4.2460	4.2091	4.1482	3.7662(3.8559)	3.8228
PI [2]↓	7.6017	7.7824	6.8942	5.9856	4.2644	4.2492	4.2648	3.8593(3.9163)	3.8438
BRISQUE [29]↓	54.899	54.988	51.563	50.936	30.213	32.103	30.542	29.030(30.1201)	28.5566

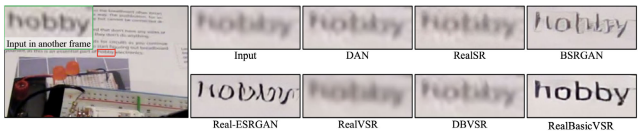
*Values in parentheses indicate the result calculated using their official code [7] and checkpoint under the same setting as RealPixVSR.



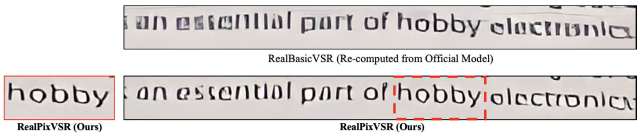
(a) Referenced from top row of [4] Figure 11.



(b) RealPixVSR result with on clip 43, frame 99.



(c) Referenced from the bottom row of [4] Figure 11.



(d) RealPixVSR result on clip 25, frame 50.

Figure 3. **Qualitative evaluation on VideoLQ [4]**. The proposed RealPixVSR effectively uses the pixel-level degradation representation and its recurrent propagation to super-resolve images with more detail compared to other methods. (a) and (c) Quantitative result referenced from the current SOTA model [4] Figure 11 top and bottom, respectively, for a direct comparison. (b) [clip-43, frame-99]. To compare how other buildings in the image are restored we extended the view to the right. RealPixVSR restores better the details of the building and windows (pointed by the yellow box) compared to RealBasicVSR. (d)[clip-25, frame-50] To compare how neighboring texts are restored we extended the view around the word *"hobby"* for comparison. RealPixVSR restores the sentence in a way that is more readable than RealBasicVSR. More comparisons can be found on supplementary material.

achieve better result with a smaller computational load of less than 80%.

5.2.2 Qualitative Comparison

Fig. 3 shows the qualitative comparison of RealPixVSR with other methods on the VideoLQ dataset. We referenced the super-resolution results of existing models from ([4] - Figure 11). We chose to borrow their results to focus more on comparing our results with RealBasicVSR, which is the current SOTA model. Fig. 3(b) shows the super-resolution result on frame 99 of the 43rd clip. To compare how other buildings in the image are restored, we have extended the view around the selected building for comparison. If we look at the buildings, we can see that RealPixVSR restores the windows and other details better than RealBasicVSR. As another example, in Fig. 3(d) we show the result of frame 50 of the 25th clip. To compare how other text in the image is restored, we have extended the view to include the word "hobby". RealPixVSR restores the sentence in a more readable way than RealBasicVSR.

Fig. 4 shows the qualitative comparison of our model with RealBasicVSR [4] on the UDM10 [50] dataset. We found that RealBasicVSR eliminates many details in images when reconstructing LR videos. It eliminates details on the wall (Fig. 4(a)), the waves on the lake (Fig. 4(b)), the grain on the wood (Fig. 4(c)), and the folds on the football jersey (Fig. 4(d)). this reduction in detail could be considered a characteristics of RealBasicVSR, and a smoother image might got a higher IQA score. On the other hand, RealPixVSR tries to preserve as much detail as possible. We believe that the pixel-level degradation representation encoder provides sufficient sensitivity to preserve detail rather than smoothing it.

5.2.3 Ablation Study

In this section, we verify the effectiveness of the proposed methods. Three different configurations are considered to distinguish the impact of each approach. The first configuration tests the presence of PDIB. This experiment is intended to verify the effectiveness of PDIB. Conventional residual blocks are used in the experiment without PDIB. The second configuration relates to the use of pixel-level

IQA	RealBasicVSR mean ± std	RealPixVSR mean ± std
NRQM [27]↑	6.631 ± 0.056	**6.843** ± 0.036
NIQE [30]]↓	4.1275 ± 0.065	**4.0056** ± 0.044
PI [2]↓	3.7483 ± 0.057	**3.5813** ± 0.035
BRISQUE [29]↓	29.3493 ± 0.946	**27.1635** ± 0.614
GFLOPs/Frame	26.5	20.5
Params(M)	6.3	6.9

Table 2. **Quantitative evaluation on UDM10 [50]** A quantitative comparison using no-reference image quality assessment(IQA) metrics is shown. RealPixVSR achieves better performance on all metrics. In terms of computational complexity, our model requires fewer FLOPs than RealBasicVSR for processing a frame. The std value denotes the standard deviation of IQA values among the 32 frames of 10 video clips in UDM10.

	(a)	(b)	(c)	(d)	(e)
Repr.	None	x_t	z_t	z_t	z_t
PDIB		✓	✓		✓
$\mathcal{L}_{pixel}$				✓	✓
NIQE↓	7.0790	7.2767	7.0307	6.9968	**6.9303**

Table 3. **Ablation results** *Repr.* indicates the pixel-level representation used for PDICN and PDIVSR. Contrastive loss is not considered for (a) and (b) as z_t is not used. (a) RealBasicVSR. None of the pixel-level representation nor PDIB is applied. (b) x_t is injected into PDIB. (c) z_t is injected into PDIB but z_t is trained without $\mathcal{L}_{pixel}$ (d) z_t is injected into resblock (e) RealPixVSR

representation z_t. In alternate experiments, z_t is substituted with the original image, x_t. The final experiment investigates the effectiveness of the pixel-level contrastive loss, $\mathcal{L}_{pixel}$, while the alternate experiment uses only $\mathcal{L}_{img}$.

Configuration We calculate the NIQE [30] scores on the UDM10 [50] dataset and make modifications to facilitate the ablation study. For each setting, we averaged the results of the 3 runs of training phase 1. The cleaning and propagation layers are reduced to 10 PDIB or Residual blocks, and the input feed is reduced to 20 frames. To ensure a fair comparison, we have increased the number of blocks to align with comparable model sizes of other configurations for Tab. 3(a). Each training session lasts 100K iterations with a batch size of 4 and a learning rate of 2.5×10^{-5}. Other configurations remain unchanged.

Effectiveness of PDIB Comparing Tab. 3(d) and 3(e), it is observed that the utilization of PDIB performs better than using conventional residual blocks. This indicates the effectiveness of the proposed PDIB. However, in Tab. 3(a) and 3(b), using PDIB with the original input x_t resulted in even worse performance than not using PDIB. This suggests that

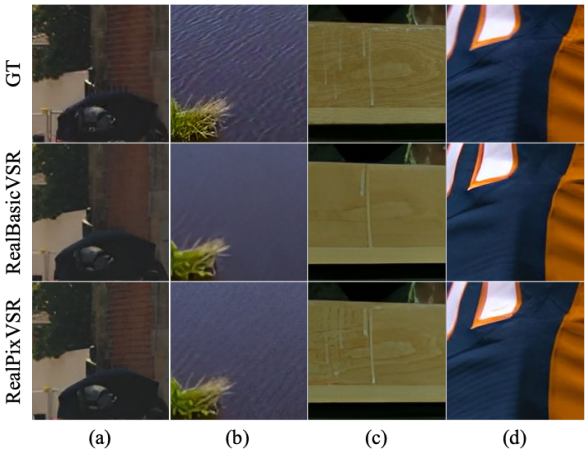


Figure 4. **Qualitative evaluation on UDM10 [50] dataset** This figure displays the qualitative comparison on the UDM10 dataset. The top row depicts the ground-truth images. The second and third rows show the results of RealBasicVSR and RealPixVSR, respectively. The proposed model reconstructs more details compared to RealBasicVSR. The displayed images are (a)archpeople, (b)lake, (c)caffe, and (d)clap.

PDIB is more compatible with z_t. This finding is reasonable as PDIB is designed to make good use of pixel-level representations. Furthermore, Tab. 3(b) and 3(d) confirms this claim by showing that replacing x_t with z_t improves performance.

Quality of Pixel-level Representation The performance can be influenced by the quality of the pixel-level representation. When comparing Tab. 3(c) and 3(e), we see significant performance enhancement with the inclusion of pixel-level contrastive loss $\mathcal{L}_{pixel}$. This finding implies that $\mathcal{L}_{pixel}$ contributes to learning a more effective pixel-level representation for reconstructing HR images. Looking at the findings presented in Tab. 3(c) and 3(d), it is evident that the quality of z_t significantly impacts the network performance, even without the influence of PDIB. This highlights the effectiveness of pixel-level contrastive loss $\mathcal{L}_{pixel}$.

6. Conclusion

In this study, we introduce a model for super-resolution of real-world videos that utilizes contrastive learning to incorporate pixel-level degradation representation. This pixel-level degradation representation seamlessly integrates with the pixel degradation-informed cleaning and VSR networks, enabling it to effectively handle complex degradation in real images. Our experiments on the UDM10 and VideoLQ datasets demonstrate that our network performs better than the state-of-the-art models.

References

[1] Sefi Bell-Kligler, Assaf Shocher, and Michal Irani. Blind super-resolution kernel estimation using an internal-gan. *CoRR*, abs/1909.06581, 2019. 1

[2] Yochai Blau, Roey Mechrez, Radu Timofte, Tomer Michaeli, and Lihi Zelnik-Manor. The 2018 PIRM challenge on perceptual image super-resolution. In *Computer Vision - ECCV 2018 Workshops*, 2018. 6, 7, 8

[3] Kelvin CK Chan, Xintao Wang, Ke Yu, Chao Dong, and Chen Change 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, 3, 5

[4] Kelvin C.K. Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. Investigating tradeoffs in real-world video super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 1, 3, 5, 6, 7

[5] Pierre Charbonnier, Laure Blanc-Féraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *ICIP*, pages 168–172, 1994. 5

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, 2020. 3

[7] Kelvin C.K.Chan. Realbasicvsr. https://github.com/ckkelvinchan/RealBasicVSR, 2022. 6, 7

[8] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 764–773, 2017. 5

[9] Dosovitskiy, Alexey, Springenberg, Jost Tobias, Riedmiller, Martin, and Brox Thomas. Discriminative unsupervised feature learning with convolutional neural networks. In *CVPR*, 2015. 3

[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*, 2014. 5

[11] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, 2020. 6

[12] Jinjin Gu, Hannan Lu, Wangmeng Zuo, and Chao Dong. Blind super-resolution with iterative kernel correction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2

[13] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita. Deep backprojection networks for super-resolution. In *CVPR*, 2018. 2

[14] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita. Recurrent back-projection network for video super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 3

[15] Jingwen He, Chao Dong, and Yu Qiao. Modulating image restoration with continual levels via adaptive feature modification layers. In *CVPR*, 2019. 2

[16] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3

[17] Olivier J. Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, S. M. Ali Eslami, and Aaron Van Den Oord. Data-efficient image recognition with contrastive predictive coding. In *Proceedings of the 37th International Conference on Machine Learning*, 2020. 3

[18] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *CVPR*, 2015. 2

[19] Allan Jabri, Andrew Owens, and Alexei Efros. Space-time correspondence as a contrastive random walk. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, 2020. 3

[20] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 2016. 5

[21] Guoliang Kang, Yunchao Wei, Yi Yang, Yueting Zhuang, and Alexander G Hauptmann. Pixel-level cycle association: A new perspective for domain adaptive semantic segmentation. In *NeurIPS*, 2020. 3

[22] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR*, 2015. 6

[23] T. Kohler, M. Batz, F. Naderi, A. Kaup, A. Maier, and C. Riess. Toward bridging the simulated-to-real gap: Benchmarking superresolution on real data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(11):2944–2959, 2020. 2

[24] Boyun Li, Xiao Liu, Peng Hu, Zhongqin Wu, Jiancheng Lv, and Xi Peng. All-in-one image restoration for unknown corruption. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17452–17462, June 2022. 1, 3, 5

[25] Xueting Li, Sifei Liu, Shalini De Mello, Xiaolong Wang, Jan Kautz, and Ming-Hsuan Yang. Joint-task self-supervised learning for temporal correspondence. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, 2019. 3

[26] Zhengxiong Luo, Yan Huang, Shang Li, Liang Wang, and Tieniu Tan. Unfolding the alternating optimization for blind super resolution. In *NeurIPS*, 2020. 1

[27] Chao Ma, Chih-Yuan Yang, Xiaokang Yang, and Ming-Hsuan Yang. Learning a no-reference quality metric for single-image super-resolution. *Comput. Vis. Image Underst.*, 158:1–16, 2017. 6, 7, 8

[28] Tomer Michaeli and Michal Irani. Nonparametric blind super-resolution. In *IEEE International Conference on Computer Vision, ICCV*, 2013. 1

[29] 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, 7, 8

[30] Anish Mittal, Rajiv Soundararajan, and Alan C. Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal Process. Lett.*, 20(3):209–212, 2013. 6, 7, 8

[31] 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*, June 2019. 6

[32] Anurag Ranjan and Michael J. Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 3

[33] Mehdi S. M. Sajjadi, Raviteja Vemulapalli, and Matthew Brown. Frame-Recurrent Video Super-Resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 3

[34] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ECCV*, 2020. 3

[35] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision – ECCV 2020: Proceedings, Part XI*, page 776–794, 2020. 3

[36] Yapeng Tian, Yulun Zhang, Yun Fu, and Chenliang Xu. Tdan: Temporally-deformable alignment network for video super-resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 3

[37] 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 (CVPR)*, June 2020. 3

[38] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang. NTIRE 2017 challenge on single image super-resolution: Methods and results. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017. 2

[39] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018. 3

[40] Longguang Wang, Yingqian Wang, Xiaoyu Dong, Qingyu Xu, Jungang Yang, Wei An, and Yulan Guo. Unsupervised degradation representation learning for blind super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10581–10590, June 2021. 1, 2, 3

[41] Xintao Wang, Kelvin C.K. Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. 1, 3

[42] Xiaolong Wang, Allan Jabri, and Alexei A. Efros. Learning correspondence from the cycle-consistency of time. In *CVPR*, 2019. 3

[43] Xinya Wang, Jiayi Ma, and Junjun Jiang. Contrastive learning for blind super-resolution via a distortion-specific network. *IEEE/CAA Journal of Automatica Sinica*, pages 1–12, 2022. 1, 3

[44] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 1905–1914, October 2021. 3, 6

[45] Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 5

[46] Gang Wu, Junjun Jiang, Xianming Liu, and Jiayi Ma. A practical contrastive learning framework for single image super-resolution. *CoRR*, abs/2111.13924, 2021. 1, 3

[47] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3

[48] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16684–16693, June 2021. 3, 4, 6

[49] Mang Ye, Xu Zhang, Pong C. Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3

[50] 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 *IEEE International Conference on Computer Vision (ICCV)*, pages 3106–3115, 2019. 6, 7, 8

[51] Jiahui Zhang, Shijian Lu, Fangneng Zhan, and Yingchen Yu. Blind image super-resolution via contrastive representation learning. *CoRR*, abs/2107.00708, 2021. 1, 3

[52] Kai Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. Designing a practical degradation model for deep blind image super-resolution. In *ICCV*, 2021. 2

[53] Kai Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. Designing a practical degradation model for deep blind image super-resolution. In *IEEE International Conference on Computer Vision*, pages 4791–4800, 2021. 1

[54] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Learning a single convolutional super-resolution network for multiple degradations. In *CVPR*, 2018. 2

[55] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 3