Contrastive Learning for Controllable Blind Video Restoration

Givi Meishvili[1]
gmeishvili@microsoft.com
Abdelaziz Djelouah[2]
abdelaziz.djelouah@disney.com
Sally Hattori[3]
sally.hattori@disney.com
Christopher Schroers[2]
christopher.schroers@disney.com

[1] Microsoft
(Work was done at DisneyResearch|Studios, before joining Microsoft)

[2] DisneyResearch|Studios
Zurich, Switzerland

[3] The Walt Disney Company
Los Angles, USA

Abstract

A lot of progress has been made since the first neural network models were trained for specific image restoration tasks, such as super-resolution and denoising. Recently multi-degradation models have been proposed, allowing for user control of the restoration process needed for real-world applications. However, this aspect is most powerful if the initial restoration can be done as best as possible in a blind setting. In parallel to this line of work, other methods can target the blind setting where, for example, in the case of super-resolution, the blur kernel is estimated for conditioning the restoration part. In particular, discriminative learning has played a key role in pushing the state of the art. Still, the learned representation cannot be interpreted or manipulated and remains a black box that doesn't offer any possibility for user-guided correction. This work addresses those issues through a representation learning pipeline that helps separate content from degradation by reasoning on pairs of degraded patches. The degradation representation is used as conditioning for a video restoration model that can denoise and upscale to arbitrary resolutions and remove film scratches. Finally, the learned representation can be mutated to fine-tune the restoration results. We demonstrate state-of-the-art results compared to the most recent video super-resolution and denoising methods.

1 Introduction

With the development of video streaming services and the increased competition between the different providers in terms of catalog size, there is a regain of interest for the studios to remaster old shows and productions to make them available on their streaming platform. Our work addresses the problem of video restoration in the context of remastering legacy video content. This content is often available in noisy, blurry, and low-resolution format and may contain scratches. Therefore, the remastering process has to address a combination of degradations and, importantly, allow for user control to have a fine-level control on the output quality. Recent developments in deep learning have pushed the state-of-the-art

Figure 1: **Controllable Blind Video Restoration.** Given a low-resolution and degraded input video, our model can be used to denoise and/or upscale. We automatically estimate the degradation present in the image (column E_d). It is possible to manipulate the degradation representation to control the restoration result and for example increase/reduce sharpness.

in the sub-problems independently by exploring different architectures or training settings in super-resolution [6, 28, 29, 50, 52, 65] and video denoising [45, 59]. However, chaining these specialized models is sub-optimal and multi-degradation models have been successfully proposed [17]. One essential requirement for adopting a restoration model by the industry is user control of the restoration process. This appears more clearly in recent works such as [11], where image sharpness can be fine-tuned locally. More recently, Kim *et al* [25] further reduce the complexity of multi-degradation models through architecture search, with the objective of interactive control of the restoration results. This requirement for user control is most powerful if the initial restoration can be done as best as possible in a blind setting, limiting user efforts to minimal fine-tuning. Blind restoration is not possible with these existing models [17, 25] which require providing the degradation parameters. In parallel, most recent blind restoration methods [49] achieve good results. However, the learned representation cannot be interpreted or manipulated.

Here we propose a multi-degradation restoration model that can address jointly denoising, super-resolution and scratch-removal. The restoration can operate in blind settings while still allowing for manipulating the result: The input video can be in low-resolution and may contain scratches. The model automatically estimates the degradation and produces restored frames both in low-resolution and high-resolution. Additionally, the output can be further manipulated to increase sharpness (see Fig. 1). A brief overview of our method during inference is presented in Figure 2. It consists of three main steps: *(i)* extracting an interpretable and controllable representation of different degradations; *(ii)* manipulating the degradations if necessary; *(iii)* finally conditioning the restoration backbone with estimated/manipulated degradation embedding. To the best of our knowledge, no solution considers the complete problem of video restoration that takes into account: Scratch removal, denoising, and upscaling while offering flexibility in manually fine-tuning the restoration of the signal.

Our training strategy leverages contrastive learning to learn an abstract representation that distinguishes various degradations in the representation space rather than explicit estimation in the pixel space. A key difference from Wang *et al*. [49] is the possibility of controlling the restoration process via manipulating the degradation features. This requires better estimates for the degradation parameters, which is possible thanks to our training strategy using pairs of degraded training samples and hard negative samples. Finally, we

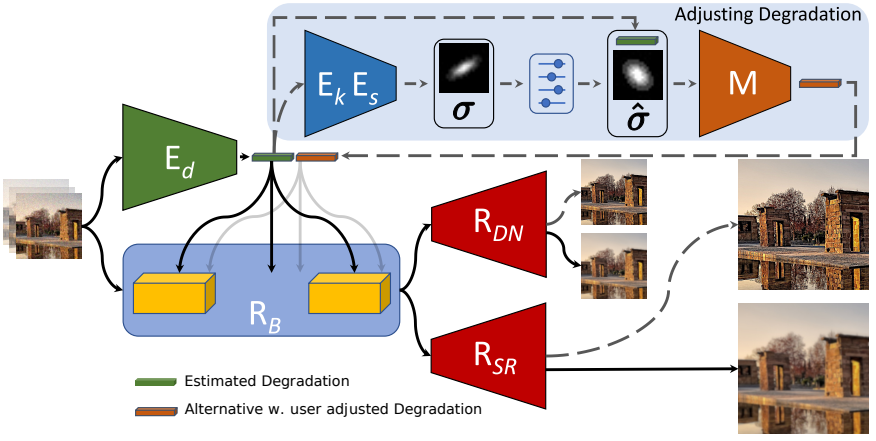

Figure 2: **Overview of our controllable restoration pipeline.** We first estimate the degradation feature by feeding the corrupted video to the encoder E_d. The degradation feature is used as conditioning for the restoration backbone R_B. It is possible to adjust both the denoising strength and blur kernel: This mutated version of the embedding (in orange) can similarly be used as conditioning for the restoration. It corresponds to the alternative outputs indicated by the dotted arrows (see text for details).

consider a wider range of degradations and address video restoration in a general setting where super-resolution is not limited to a discrete set of scaling factors is necessary when processing video formats like NTSC. Our contributions can be summarized as follows: *(i)* A video restoration model that can jointly address multiple types of degradations. *(ii)* A new contrastive training strategy to learn an interpretable and controllable representation of different degradations. *(iii)* State-of-the-art results in blind video restoration.

2 Related Work

Super-Resolution. This research area is active and has primarily benefited from the latest advances in deep learning (see, *e.g.*, [13, 26, 28, 52]). An important part of super-resolution research works has focused on improving task-specific CNN architectures and components (see *e.g.*, [1, 12, 27, 29, 30, 33, 38, 43, 50, 55, 57, 60, 64, 65, 66]). Many other aspects have been considered, ranging from using adversarial training for realistic detail hallucination [4, 28, 37, 62]), to improve the realism of the training set through accurate modeling [7, 54] or through using real zoomed-in images [8, 63]. Temporal information can also be used in the context of video super-resolution [6, 16, 22, 24, 31, 44, 47, 51, 56].

However, in this work, we focus more on methods addressing blind super-resolution [2, 11, 15, 34, 40]. These rely on some form of *test-time* optimization to estimate the blur kernel and predict the corresponding high-resolution output. These two steps can be done separately [34], jointly [11, 15] or require a fine-tuning of the super-resolution model [2, 40]. In the case of blind video super-resolution, Pan *et al*. [36] estimate a blur kernel used in an image deconvolution step. The resulting image is then restored using a neural network and aligned adjacent frames. We can note that this strategy may not be optimal as the restoration neural network cannot directly leverage the blur kernel information.

Denoising. Similarly to super-resolution, a lot of progress has been made since early works based on neural networks [5, 21, 53]. We focus here on recent video denoising methods:

Yue *et al*. [59] proposed a raw video denoising network (RViDeNet) by exploring the temporal, spatial, and channel correlations of video frames. Tassano *et al*. [46] proposed a video denoising algorithm based on a convolutional neural network model conditioned on the noise level. Maggioni *et al*. [32] introduced an a multi-stage algorithm to reduce the complexity while maintaining denoising performance. These methods strongly rely on providing noise level as input. Closer to our blind setting, Claus *et al*. [10] use a multi-frame neural network architecture to denoise videos and considered varied noise models during training. Although more robust than specialized denoisers, results are not competitive with more recent methods leveraging noise parameters at test time.

Scratch Removal. Scratch removal is a classical mixed degradation problem when working with old photo/video data, and most existing methods consider it an image inpainting problem [3, 8, 14, 41]. Some works consider joint restoration of images corrupted by a combination of different distortions [42, 58]. Wan *et al*. [48] proposed a triplet domain translation network by leveraging real photos and synthetic image pairs and trained two variational autoencoders (VAEs) to transform old photos and clean photos into two latent spaces. And the translation between these two latent spaces is learned with synthetic paired data.

Multi-degradation models. Combining multiple specialized models to restore images is not optimal and efficient conditioning for a multi-degradation model was proposed by He*et. al* [17]. This model was recently used by Wang *et. al* [49] in the blind restoration setting, leveraging contrastive learning to avoid *test-time optimization* while still conditioning the restoration model on the estimated degradation. However, the proposed model is limited to images, fixed scaling factors and the learned representation cannot be interpreted or manipulated. Finally, Kim *et al* [25] further reduce the complexity of multi-degradation models through architecture search, with the objective of interactive control of the restoration results. They still require providing the degradation parameters.

3 Method

We aim to build a model that can restore videos corrupted by the most common set of degradation present in legacy film content, namely: scratches, noise, and the implicit blur in the low-resolution input. We can briefly formulate the degradation model of a set of consecutive low-resolution (LR) frames y as follows:

$$y = S \circ \Big((x * k) \downarrow_s + n \Big) \tag{1}$$

where x is the corresponding unknown set of consecutive high-resolution (HR) frames, $*$ is convolution operation, k is a blur kernel, $\downarrow_s$ denotes downsampling operation by factor s, n stands for noise, and S represents a film scratch as a mask that sets pixel color values to 1.

As illustrated in Figure 2, we train an encoder E_d capable of extracting a latent representation for the degradation present in the input set of frames. For this, we leverage recent advances in contrastive learning [18, 49]. This latent representation is then used as conditioning for the feature restoration backbone R_B, which is used both for low-resolution denoising, with R_{DN}, and the super-resolution path R_{SR}. We leverage information from multiple frames in our model without using explicit motion estimation - a strategy already successfully used in frame interpolation [23]. We use a set of 5 input frames for each output frame: the current frame and 2 temporally adjacent frames from the past and future. Our model also learns to decode the degradation representation into blur kernel and noise levels. Furthermore, it is possible to modify these parameters and adjust the latent representation accordingly, thanks

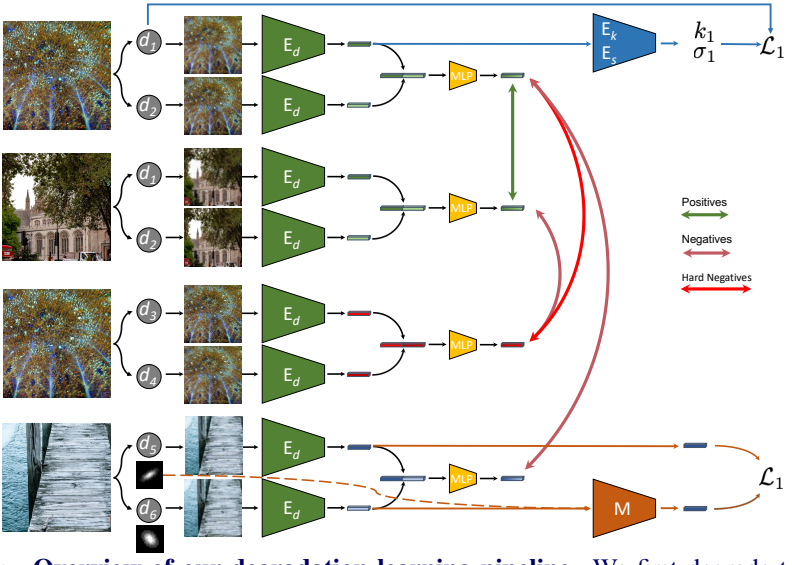

Figure 3: **Overview of our degradation learning pipeline.** We first degrade two high-resolution input images with a pair of degradations d_1, d_2. We encode low-resolution degraded image pairs using encoder E_d. Later, features of the first and second rows are concatenated and passed to a two-layer MLP network. Final outputs connected with a green arrow form a positive pair for contrastive learning. A red feature from the third row creates a hard negative example for the feature from the first row since its obtained via encoding the same image corrupted with degradations d_3 and d_4. We additionally regress the blur kernel k_1 and noise level σ_1 via encoders E_k and E_s, respectively. We also learn to manipulate features using encoder M by supplying it with adjusted degradation parameters k_5, σ_5 and obtain $z_p^5 = M(z_p^6, k_5, \sigma_5)$.

to the mutator model M. This flexibility is needed in real-world applications where artists may want to control sharpness levels and denoising strength.

In the following, we first learn video degradation representation, then our proposal to allow the manipulation of the learned latent representation and finally, we take advantage of the learned representation to condition the restoration task.

Video Degradation Representation The objective is to learn to extract from the input frames a latent representation that should be discriminative towards different degradations in the input. More precisely, two different, similarly degraded videos should lead to two embeddings close to each other. In contrast, the two differently degraded versions of the same video should result in latent representations further apart. This is a more challenging objective than the one considered by Wang *et al.* [49], which is a more straightforward application of the Moco [18] representation learning framework: the loss was designed such as to push further away the embedding of patches from different images while bringing closer patches from the same image. Such an objective doesn't encourage a clear disentanglement between the content and the degradation.

We are interested in disentangling the degradation from the content, but different samples from the training set are captured with sensors of varying resolutions, exposures, and noise levels. Any high-resolution image already contains a certain amount of degradation, and the application of the degradation model from Equation 1 will result in a mixture of two degra-

dations: inherent from a high-resolution image and one from equation 1. Separating these two degradations is an ill-posed problem. Therefore, directly training the encoder E_d with a Multilayer Perceptron (MLP) that tries to optimize our contrastive learning objective is not optimal. To address this issue, we propose to train the encoder E_d using pairs of degraded patches obtained from sampling a random high-resolution image and degrading it with two different degradations. Consequently, the MLP should focus on differences between degradations introduced during training rather than the ones present in the original high-resolution video.

An overview of the training procedure is presented in Figure 3. Let us denote a specific set of different degradations from equation 1 as $d_i \sim \mathcal{D}$ parameterized by blur kernel k_i and noise level σ_i, $y_p^i = d_i(x_p)$ as video x_p degraded with degradation d_i, and $z_p^i = E_d(y_p^i)$ as latent vector obtained by encoding y_p^i using encoder E_d. We sample pairs of degradations (d_i, d_j), (d_k, d_l), and videos x_p, x_q. We apply pairs of sampled degradations to the videos and encode them using encoder E_d: $x_p \rightarrow \Big(d_i(x_p), d_j(x_p)\Big) \rightarrow (y_p^i, y_p^j) \rightarrow (z_p^i, z_p^j)$

$x_q \rightarrow \Big(d_i(x_q), d_j(x_q)\Big) \rightarrow (y_q^i, y_q^j) \rightarrow (z_q^i, z_q^j) \quad x_p \rightarrow \quad \Big(d_k(x_p), d_l(x_p)\Big) \rightarrow (y_p^k, y_p^l) \rightarrow (z_p^k, z_p^l)$

where superscripts and subscripts denote degradations and input videos, respectively. Note that embedding pairs (z_p^i, z_p^j) and (z_q^i, z_q^j) are obtained by degrading two different videos x_p and x_q, with the same pair of degradations (d_i, d_j). Therefore, they form a positive pair. Hard negative pairs (z_p^i, z_p^j) and (z_p^k, z_p^l) are obtained by degrading the same video x_p with different pairs of degradations: (d_i, d_j) and (d_k, d_l). We provide these difficult negative examples during training to force the neural representation to focus on the degradation rather than the content. Next, we define the relative degradations via concatenating the resulting embedding pairs and following the Moco framework feed them to a two-layer MLP projection head F: $\psi_p^{ij} = F\Big([z_p^i, z_p^j]\Big)$, $\psi_q^{ij} = F\Big([z_q^i, z_q^j]\Big)$, and $\psi_p^{kl} = F\Big([z_p^k, z_p^l]\Big)$. We want ψ_p^{ij} to be similar to ψ_q^{ij} since they share the same relative degradations and are dissimilar to ψ_p^{kl} since degradations are different. Therefore, an InfoNCE loss is used to measure the similarity:

$$\mathcal{L}_c = \sum_{p,q}^{\mathcal{V}} \sum_{i,j}^{\mathcal{D}} -\log \frac{e^{\left(\psi_p^{ij} \cdot \psi_q^{ij} / \tau\right)}}{\sum_{t=1}^{N_Q} e^{\left(\psi_p^{ij} \cdot \psi_t / \tau\right)} + e^{\left(\psi_p^{ij} \cdot \psi_p^{kl} / \tau\right)}} \tag{2}$$

where a different degradation pair kl is randomly sampled for each degradation pair ij. N_Q is the number of samples in the MoCo queue, $\mathcal{V}$ is a set of training videos, $\mathcal{D}$ is a set of degradations, τ is a temperature parameter, and $\cdot$ denotes the dot product between two vectors.

To allow the modification of the results and fine-tuning of the outputs in addition to optimizing for $\mathcal{L}_c$ we also estimate the parameters k_i and σ_i of applied degradation d_i. We train a small degradation regressor MLPs: E_k and E_s that regress the parameters k_i and σ_i, in a standardized format by optimizing:

$$\mathcal{L}_k = \sum_{p}^{\mathcal{V}} \sum_{i}^{\mathcal{D}} \left| E_k\Big(E_d\Big(d_i(x_p)\Big)\Big) - k_i \right| \qquad \mathcal{L}_s = \sum_{p}^{\mathcal{V}} \sum_{i}^{\mathcal{D}} \left| E_s\Big(E_d\Big(d_i(x_p)\Big)\Big) - \sigma_i \right| \tag{3}$$

where subscripts k and σ identify the specific output of the model E.

Overall training objective can be summarized as follows:

$$\mathcal{L} = \lambda_c \mathcal{L}_c + \lambda_k \mathcal{L}_k + \lambda_s \mathcal{L}_s \tag{4}$$

Learning to Manipulate Degradations. Our goal is to restore the distorted videos. However, we also want to have fine-grained control over this process. For example, one might

need to correct the blur kernel, adjust the noise level and obtain the alternatively restored video. Therefore, we freeze the pre-trained encoder E_d and train the model M to perform manipulations in the latent space of degradations. Given the embedding $z_p^i = E_d\Big(d_i(x_p)\Big)$ and some new adjusted parameters k_j, σ_j, the model M enables the manipulations in the latent space and regresses the feature $z_p^j = M(z_p^i, k_j, \sigma_j)$. During training we sample video x_p, and a pair of degradations: d_i, d_j. Next, we degrade x_p obtaining $y_p^i = d_i(x_p)$ and $y_p^j = d_j(x_p)$. We compute encodings $z_p^i = E_d(y_p^i), z_p^j = E_d(y_p^j)$ using frozen encoder E_d. Finally, we train model M by minimizing the following objective:

$$\mathcal{L}_m = \sum_p^{\mathcal{V}} \sum_{i,j}^{\mathcal{D}} \left| M(z_p^i, k_j, \sigma_j) - z_p^j \right| \tag{5}$$

Learning Conditional Restoration. As illustrated in Figure 2, the proposed model extracts from consecutive frames an encoding of the degradation present in the video. This degradation, expressed as a latent vector, is then used as conditioning for the restoration. Formally our model consists of restoration backbone R_B and two task-specific branches: R_{SR} and R_{DN} for super-resolution and denoising, respectively. The motivation for having a shared backbone R_B is to simultaneously learn features beneficial for different restoration tasks. While the networks R_{SR} and R_{DN} should learn features tailored for super-resolution, denoising, and scratch removal, respectively.

Given a corrupted input y_p^i we first obtain the corresponding degradation embedding $E_d(y_p^i)$. We pass both y_p^i and $E_d(y_p^i)$ to the restoration backbone R_B. Consequently, the resulting final feature map from R_B is fed to R_{SR} and R_{DN} subnetworks, respectively. Therefore, we produce two outputs in this model. The first is the low-resolution denoised image and, consequently, the original low-resolution noise. The second is the denoised high-resolution image. Rather than outputting a fixed $4\times$ super-resolved frame, we employ Meta Upscale module [19] at the end of our R_{SR} model to enable non-integer upsampling factors and address more general scenarios. Additionally, our model must remove the possible scratches presented in the video for both super-resolution and denoising branches. Hence in addition to the losses mentioned in the equation 4, during training models R_{SR} and R_{DN} are trained to minimize objectives $\mathcal{L}_{SR}$ and $\mathcal{L}_{DN}$ respectively.

$$\mathcal{L}_{SR} = \sum_p^{\mathcal{V}} \sum_i^{\mathcal{D}} \left| R_{SR}\Big(E_d(y_p^i), y_p^i\Big) - \hat{x}_p \right| \qquad \mathcal{L}_{DN} = \sum_p^{\mathcal{V}} \sum_i^{\mathcal{D}} \left| R_{DN}\Big(E_d(y_p^i), y_p^i\Big) - (\hat{x}_p * k_i)\downarrow_s \right| \tag{6}$$

where $\hat{x}_p$ corresponds to the middle high-resolution ground-truth frame of the set of frames x_p. In addition to the content losses mentioned in equations 6, we also keep fine-tuning the degradation encoder and manipulation models. Therefore, our final objective becomes:

$$\mathcal{L} = \lambda_{SR}\mathcal{L}_{SR} + \lambda_{DN}\mathcal{L}_{DN} + \lambda_c\mathcal{L}_c + \lambda_k\mathcal{L}_k + \lambda_s\mathcal{L}_s \tag{7}$$

4 Experiments

We incorporated the Vid4 and Set8 datasets for comparison and ablation purposes. We generated multiple degraded versions of the original datasets to demonstrate the capabilities of our pipeline in different settings. First, we created multiple blurry versions of each dataset using nine blur kernels presented in Table 2. Afterward, we downsampled and corrupted each blurry dataset using AWGN of different magnitudes. And finally, we followed the Wan

Feature Contrasting	MAE↓	Kernel Similarity↑
Single	0.0008	0.9438
KernelGAN[2]	0.0006	0.9446
Pairwise	**0.0005**	**0.9821**

Table 1: Kernel estimation accuracy for single and pairwise feature contrasting strategies. The first and second rows correspond to single and pairwise feature contrasting strategies, respectively. Results obtained using the KernelGAN[2] are reported in the second row. We report Mean Absolute Error, and Kernel Similarity [20].

et al. [48] method to generate scratched versions of the datasets. For quantitative comparison, we used PSNR and SSIM. Additional training and implementation details can be found in the supplementary material.

Single vs Pairwise Contrasting Ablation. An essential pipeline component is the encoder E_d that learns to map the degraded videos to the latent space. As we mentioned previously, features from the latent space should reflect as much information as possible about the degradation in the input video. Therefore, we evaluate the latent space via the quality of the blur kernels that the encoder E_k produces given a feature from E_d as input. We use Kernel Similarity [20] and Mean Absolute Error (MAE) as evaluation metrics between ground truth and estimated kernels. We thus perform ablation experiments for different ways to train the encoder E_d and justify the choice of the pairwise training strategy. We consider two possible design choices: *(i)* training E_d by contrasting single video embeddings, and *(ii)* training E_d by contrasting pairs of video embeddings. MAE's and Kernel Similarities are reported in Table 1. One can observe that the Pairwise feature contrasting strategy leads to a better quality of the estimated kernels.

Initial vs Mutated Kernels Ablation. We also evaluated how well mutator M manipulates the latent input features. Specifically, we are interested in consistency between the input kernel to the model M and kernel-related information contained in the output manipulated latent code. Towards this goal, we first feed model M with a latent code, noise level, and adjusted blur kernel; and obtain the adjusted code. After, we provide the modified latent code to the Kernel estimator E_k and estimate the adjusted kernel. Finally, we measure the MAE and Kernel Similarity between the initial adjusted blur kernel and the estimated blur kernel after manipulation. We performed the mentioned procedure on degraded videos from Set8 and obtained $MAE = 0.0004$ and $KS = 0.9837$ (Kernel Similarity).

Video Super-Resolution Comparisons. We performed a quantitative comparison with the non-blind video super-resolution approach of Tian *et al*. [47], and with the blind methods of Pan *et al*. [35], and Zhang *et al*. [61]. We report results for different blur kernels and noise levels in Table 2. It provides mean PSNR/SSIM per kernel and per-noise level for different methods. This allows analyzing all cases and observing how different methods perform on isotropic/anisotropic gaussian blur kernels and noise levels of different magnitudes. Our method achieves the best performance in all settings except for the one closest to the bicubic kernel, which is the one where naturally a specialized model [47] performs best. To understand the benefits of our multi-frame and pairwise training, we retrained the model of Wang *et al*. [49] on the Vimeo90K[56] and evaluated it on the Vid4 and Set8 test sets. Retrained model achieves 22.47 / 0.63 and, 26.80 / 0.74 for Vid4 and Set8, respectively. In contrast, our model achieves 22.73 / 0.66 on Vid4 and 27.07 / 0.76 on Set8 while simultaneously addressing multiple restoration tasks, handling non-integer scaling factors, and manipulating results.

	σ	Method	Blur Kernels									All
SET8	0	Ours	27.51 / 0.82	27.17 / 0.79	24.25 / 0.67	27.76 / 0.81	26.40 / 0.77	25.52 / 0.73	27.74 / 0.81	26.44 / 0.77	25.38 / 0.72	**26.46 / 0.77**
		Tian *et al.* [43]	29.22 / 0.84	26.20 / 0.75	24.20 / 0.67	27.38 / 0.79	25.66 / 0.74	24.94 / 0.70	27.49 / 0.80	25.80 / 0.74	25.02 / 0.70	26.21 / 0.75
		Pan *et al.* [28]	26.89 / 0.79	25.76 / 0.73	23.99 / 0.66	25.82 / 0.74	24.74 / 0.70	24.66 / 0.69	26.52 / 0.77	25.02 / 0.71	24.72 / 0.69	25.35 / 0.72
		Zhang *et al.* [49]	25.80 / 0.72	24.95 / 0.70	23.95 / 0.65	25.17 / 0.71	24.09 / 0.66	24.12 / 0.66	25.38 / 0.71	24.32 / 0.67	24.22 / 0.66	24.67 / 0.68
	5	Ours	27.94 / 0.82	27.74 / 0.79	24.74 / 0.68	28.22 / 0.81	26.81 / 0.76	26.06 / 0.73	28.10 / 0.81	26.82 / 0.77	26.10 / 0.73	**26.95 / 0.77**
		Tian *et al.* [43]	29.44 / 0.83	26.51 / 0.74	24.42 / 0.66	27.71 / 0.78	25.89 / 0.73	25.18 / 0.69	27.82 / 0.79	26.06 / 0.73	25.28 / 0.69	26.48 / 0.74
		Pan *et al.* [28]	27.18 / 0.78	26.07 / 0.73	24.22 / 0.65	26.11 / 0.74	24.97 / 0.70	24.90 / 0.68	26.85 / 0.76	25.28 / 0.71	24.98 / 0.69	25.62 / 0.72
		Zhang *et al.* [49]	26.16 / 0.72	25.27 / 0.69	24.21 / 0.65	25.49 / 0.71	24.35 / 0.66	24.40 / 0.66	25.73 / 0.71	24.64 / 0.66	24.52 / 0.66	24.97 / 0.68
	10	Ours	28.51 / 0.81	28.14 / 0.78	25.26 / 0.68	28.58 / 0.80	27.13 / 0.75	26.53 / 0.73	28.52 / 0.80	27.35 / 0.76	26.72 / 0.73	**27.42 / 0.76**
		Tian *et al.* [43]	29.01 / 0.79	26.68 / 0.71	24.61 / 0.63	27.74 / 0.75	26.03 / 0.69	25.36 / 0.66	27.80 / 0.75	26.17 / 0.70	25.47 / 0.66	26.54 / 0.70
		Pan *et al.* [28]	27.32 / 0.77	26.35 / 0.71	24.45 / 0.64	26.35 / 0.73	25.21 / 0.68	25.14 / 0.67	27.06 / 0.75	25.53 / 0.70	25.23 / 0.67	25.85 / 0.70
		Zhang *et al.* [49]	26.51 / 0.72	25.61 / 0.68	24.35 / 0.63	25.86 / 0.70	24.72 / 0.65	24.69 / 0.65	26.07 / 0.70	24.99 / 0.66	24.81 / 0.65	25.29 / 0.67
	15	Ours	28.54 / 0.81	28.26 / 0.77	25.55 / 0.68	28.61 / 0.79	27.35 / 0.75	26.78 / 0.72	28.60 / 0.79	27.48 / 0.75	26.92 / 0.72	**27.57 / 0.75**
		Tian *et al.* [43]	28.17 / 0.75	26.49 / 0.67	24.58 / 0.59	27.31 / 0.70	25.87 / 0.65	25.30 / 0.62	27.35 / 0.71	25.99 / 0.66	25.40 / 0.62	26.27 / 0.66
		Pan *et al.* [28]	27.15 / 0.75	26.35 / 0.69	24.51 / 0.61	26.33 / 0.71	25.27 / 0.67	25.20 / 0.64	26.98 / 0.73	25.58 / 0.68	25.29 / 0.64	25.85 / 0.68
		Zhang *et al.* [49]	26.62 / 0.71	25.77 / 0.67	24.44 / 0.62	26.05 / 0.69	24.98 / 0.65	24.88 / 0.64	26.22 / 0.69	25.20 / 0.65	25.00 / 0.64	25.46 / 0.66
	25	Ours	27.44 / 0.79	27.52 / 0.75	25.51 / 0.67	27.63 / 0.77	26.82 / 0.73	26.48 / 0.71	27.65 / 0.78	26.88 / 0.74	26.55 / 0.71	**26.94 / 0.74**
		Tian *et al.* [43]	25.89 / 0.65	25.16 / 0.57	23.86 / 0.50	25.58 / 0.61	24.72 / 0.56	24.39 / 0.53	25.61 / 0.61	24.80 / 0.56	24.44 / 0.53	24.94 / 0.57
		Pan *et al.* [28]	25.97 / 0.70	25.44 / 0.61	24.05 / 0.54	25.52 / 0.66	24.74 / 0.62	24.61 / 0.57	25.92 / 0.66	25.00 / 0.63	24.66 / 0.57	25.10 / 0.61
		Zhang *et al.* [49]	25.96 / 0.70	25.41 / 0.65	24.30 / 0.60	25.63 / 0.67	24.85 / 0.64	24.73 / 0.62	25.72 / 0.68	24.97 / 0.64	24.79 / 0.62	25.15 / 0.65
	All	Ours	27.99 / **0.81**	**27.77 / 0.78**	**25.06 / 0.68**	**28.16 / 0.80**	**26.90 / 0.75**	**26.27 / 0.72**	**28.12 / 0.80**	**26.99 / 0.76**	**26.33 / 0.72**	**27.07 / 0.76**
		Tian *et al.* [43]	**28.35** / 0.77	26.21 / 0.69	24.33 / 0.61	27.14 / 0.73	25.63 / 0.67	25.03 / 0.64	27.21 / 0.73	25.76 / 0.68	25.12 / 0.64	26.09 / 0.68
		Pan *et al.* [28]	26.90 / 0.76	25.99 / 0.69	24.24 / 0.62	26.03 / 0.72	24.99 / 0.67	24.90 / 0.65	26.67 / 0.73	25.28 / 0.69	24.98 / 0.65	25.55 / 0.68
		Zhang *et al.* [49]	26.21 / 0.71	25.40 / 0.68	24.25 / 0.63	25.64 / 0.70	24.60 / 0.65	24.56 / 0.65	25.82 / 0.70	24.82 / 0.66	24.67 / 0.65	25.11 / 0.67
VID4	0	Ours	22.84 / 0.73	23.49 / 0.71	21.11 / 0.53	23.41 / 0.73	22.60 / 0.68	22.44 / 0.64	23.53 / 0.73	22.63 / 0.66	21.99 / 0.61	**22.67 / 0.67**
		Tian *et al.* [43]	24.62 / 0.77	22.17 / 0.62	20.79 / 0.51	23.17 / 0.69	21.86 / 0.61	21.32 / 0.55	23.09 / 0.68	21.82 / 0.60	21.32 / 0.55	22.24 / 0.62
		Pan *et al.* [28]	22.82 / 0.69	21.88 / 0.59	20.66 / 0.50	21.97 / 0.62	21.11 / 0.56	21.12 / 0.54	22.37 / 0.64	21.27 / 0.57	21.12 / 0.53	21.60 / 0.58
		Zhang *et al.* [49]	22.24 / 0.62	21.65 / 0.59	20.82 / 0.53	21.80 / 0.61	20.91 / 0.56	20.96 / 0.55	21.85 / 0.60	20.97 / 0.55	21.07 / 0.55	21.36 / 0.57
	5	Ours	23.03 / 0.74	23.59 / 0.71	21.47 / 0.55	23.60 / 0.73	22.76 / 0.67	22.55 / 0.64	23.54 / 0.73	22.65 / 0.66	22.47 / 0.63	**22.85 / 0.67**
		Tian *et al.* [43]	24.52 / 0.76	22.22 / 0.61	20.84 / 0.50	23.21 / 0.68	21.88 / 0.60	21.35 / 0.54	23.12 / 0.67	21.84 / 0.59	21.36 / 0.54	22.26 / 0.61
		Pan *et al.* [28]	22.81 / 0.69	21.94 / 0.59	20.71 / 0.49	22.03 / 0.62	21.15 / 0.56	21.16 / 0.53	22.40 / 0.64	21.31 / 0.56	21.18 / 0.53	21.63 / 0.58
		Zhang *et al.* [49]	22.29 / 0.63	21.76 / 0.59	20.96 / 0.52	21.90 / 0.61	20.99 / 0.56	21.07 / 0.55	21.91 / 0.60	21.03 / 0.55	21.16 / 0.55	21.45 / 0.57
	10	Ours	23.52 / 0.74	23.51 / 0.69	21.68 / 0.56	23.67 / 0.72	22.71 / 0.66	22.59 / 0.63	23.64 / 0.72	22.84 / 0.66	22.62 / 0.63	**22.98 / 0.67**
		Tian *et al.* [43]	24.15 / 0.72	22.21 / 0.58	20.85 / 0.47	23.09 / 0.65	21.82 / 0.57	21.39 / 0.52	22.96 / 0.64	21.82 / 0.56	21.36 / 0.51	22.18 / 0.58
		Pan *et al.* [28]	22.69 / 0.67	21.95 / 0.57	20.75 / 0.48	22.01 / 0.61	21.16 / 0.55	21.22 / 0.52	22.32 / 0.62	21.34 / 0.55	21.20 / 0.52	21.63 / 0.57
		Zhang *et al.* [49]	22.24 / 0.62	21.75 / 0.58	20.85 / 0.51	21.91 / 0.60	21.04 / 0.55	21.13 / 0.54	21.87 / 0.59	21.10 / 0.54	21.14 / 0.53	21.45 / 0.56
	15	Ours	23.36 / 0.74	23.31 / 0.68	21.69 / 0.55	23.48 / 0.72	22.73 / 0.65	22.49 / 0.62	23.56 / 0.71	22.77 / 0.65	22.49 / 0.61	**22.88 / 0.66**
		Tian *et al.* [43]	23.62 / 0.68	22.02 / 0.54	20.76 / 0.44	22.79 / 0.61	21.73 / 0.54	21.25 / 0.49	22.74 / 0.60	21.65 / 0.53	21.24 / 0.48	21.98 / 0.55
		Pan *et al.* [28]	22.51 / 0.66	21.86 / 0.55	20.72 / 0.46	21.90 / 0.60	21.18 / 0.53	21.15 / 0.50	22.26 / 0.60	21.29 / 0.54	21.15 / 0.50	21.56 / 0.55
		Zhang *et al.* [49]	22.11 / 0.61	21.62 / 0.56	20.73 / 0.49	21.82 / 0.59	21.09 / 0.54	21.04 / 0.52	21.84 / 0.58	21.07 / 0.53	21.04 / 0.52	21.37 / 0.55
	25	Ours	22.61 / 0.72	22.70 / 0.66	21.39 / 0.54	22.73 / 0.70	22.22 / 0.64	22.09 / 0.60	22.77 / 0.69	22.17 / 0.63	22.00 / 0.59	**22.30 / 0.64**
		Tian *et al.* [43]	22.33 / 0.60	21.38 / 0.47	20.35 / 0.38	21.85 / 0.54	21.09 / 0.47	20.81 / 0.42	21.78 / 0.53	21.03 / 0.46	20.73 / 0.42	21.26 / 0.48
		Pan *et al.* [28]	21.85 / 0.61	21.41 / 0.50	20.43 / 0.41	21.40 / 0.55	20.83 / 0.50	20.86 / 0.45	21.63 / 0.55	20.94 / 0.50	20.79 / 0.44	21.13 / 0.50
		Zhang *et al.* [49]	21.54 / 0.59	21.18 / 0.54	20.40 / 0.48	21.32 / 0.57	20.78 / 0.53	20.75 / 0.51	21.26 / 0.56	20.71 / 0.52	20.66 / 0.50	20.96 / 0.53
	All	Ours	23.07 / **0.73**	**23.32 / 0.69**	**21.47 / 0.55**	**23.38 / 0.72**	**22.60 / 0.66**	**22.43 / 0.63**	**23.41 / 0.72**	**22.61 / 0.65**	**22.31 / 0.61**	**22.73 / 0.66**
		Tian *et al.* [43]	**23.85** / 0.71	22.00 / 0.56	20.72 / 0.46	22.82 / 0.63	21.68 / 0.56	21.22 / 0.50	22.74 / 0.62	21.63 / 0.55	21.20 / 0.50	21.98 / 0.57
		Pan *et al.* [28]	22.54 / 0.66	21.81 / 0.56	20.65 / 0.47	21.86 / 0.60	21.09 / 0.54	21.10 / 0.51	22.20 / 0.61	21.23 / 0.54	21.09 / 0.50	21.51 / 0.56
		Zhang *et al.* [49]	22.08 / 0.61	21.59 / 0.57	20.75 / 0.51	21.75 / 0.60	20.96 / 0.55	20.99 / 0.53	21.75 / 0.59	20.98 / 0.54	21.01 / 0.53	21.32 / 0.56

Table 2: Quantitative comparison to other video super-resolution methods at 4x scaling factor. We report PSNR/SSIM values of our and competitor methods on the Set8 and Vid4 datasets. Different rows and columns correspond to different AWGN levels and blur kernels.

σ	Method	Dataset VID4	Dataset SET8	σ	Method	Dataset VID4	Dataset SET8	σ	Method	Dataset VID4	Dataset SET8
5	Ours	**40.85 / 0.99**	**40.22 / 0.99**	10	Ours	**35.46 / 0.99**	**35.09 / 0.99**	All	Ours	**34.09 / 0.99**	**33.65** / 0.98
	UDVD [23]	36.66 / 0.98	38.11 / 0.97		UDVD [23]	33.41 / 0.97	33.96 / 0.95		UDVD [23]	32.52 / 0.97	32.71 / 0.95
	DVDnet [28]	37.92 / 0.99	39.30 / 0.99		DVDnet [28]	34.27 / 0.98	34.02 / 0.96		DVDnet [28]	33.07 / 0.97	32.7 / 0.95
	FastDVDnet [40]	40.68 / 0.99	39.80 / 0.99		FastDVDnet [40]	34.83 / 0.99	34.52 / 0.99		FastDVDnet [40]	33.57 / **0.99**	33.14 / **0.99**
15	Ours	**32.30 / 0.99**	**31.78** / 0.98	25	Ours	27.76 / **0.99**	**27.50** / 0.97				
	UDVD [23]	31.52 / 0.96	31.27 / 0.94		UDVD [23]	**28.48** / 0.95	27.48 / 0.93				
	DVDnet [28]	31.94 / 0.97	30.81 / 0.94		DVDnet [28]	28.15 / 0.94	26.67 / 0.89				
	FastDVDnet [40]	31.64 / 0.99	31.24 / **0.99**		FastDVDnet [40]	27.11 / **0.99**	26.98 / **0.99**				

Table 3: Quantitative comparison to the non-blind video denoising methods. We report PSNR/SSIM values on VID4 and Set8 datasets.

Method	Dataset VID4	Dataset SET8
Ours	**36.09 / 0.99**	**31.93 / 0.98**
Wan *et al.* [48]	24.54 / 0.83	26.98 / 0.86

Table 4: Quantitative comparison to the scratch removal method of Wan *et al.* [48].

Video Denoising Comparisons. We performed a quantitative comparison with the video denoising methods of Tassano *et al.* [45, 46], and Sheth *et al.* [39]. We report results for

different noise levels in Table 3. Our blind method achieves competitive performance and slightly outperforms the model of [46] , which has access to the noise level as input.

Figure 4: **Qualitative Comparison Super-Resolution.** We performed a qualitative comparison with methods of Tian *et al*. [47] and Pan *et al*. [35]. Different rows correspond to different combinations of blur kernels and noise levels. The first column corresponds to a low-resolution input middle frame. Next, the second and third columns correspond to the restored results of Tian *et al*. [47] and Pan *et al*. [35], respectively. The fourth column shows the results of our pipeline. Finally, the last column corresponds to the ground-truth frame.

Video Scratch Removal Comparisons. We performed a quantitative comparison with the method of Wan *et al*. [48]. In this experiment, we generated corrupted versions of Vid4 and Set8 by first adding AWGN with $\sigma = 5$ and applying synthetic scratches following Wan *et al*. [48]'s protocol. We report PSNR/SSIM metrics in Table 4. Our method outperforms the competitor's method. Note that our pipeline takes scratched videos as input while [48] takes a single scratched frame. A significant performance gap can be explained by our method leveraging information from the temporal dimension, which is not available in the case of [48]. On the other hand, [48] takes the mask of the scratched region as input, which simplifies the restoration process.

Manipulating Real Videos. A real video restoration example is presented in Figure 1. One can observe the gradual decrease of the blur level in restored frames from left to the right. Initially, we pass the feature from encoder E_d to backbone R_b and obtain the results in the 4-th column. We manipulate the blur kernel to both more and less blur. We feed the mutator M with the modified blur kernels to obtain new embeddings to condition the restoration. One can see the effect from blurry to sharper results.

5 Conclusion

In this paper, we proposed a discriminative learning strategy that helps separate content from degradation by reasoning on pairs of degraded patches, where both content and degradation vary independently. The degradation representation is used as conditioning for a video restoration model that can handle denoising, super-resolution, and scratch removal. More importantly, the learned representation can be manipulated to fine-tune the results, which is crucial for real application scenarios.

References

[1] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[2] Sefi Bell-Kligler, Assaf Shocher, and Michal Irani. Blind super-resolution kernel estimation using an internal-gan. 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. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/5fd0b37cd7dbbb00f97ba6ce92bf5add-Paper.pdf.

[3] V. Bruni and D. Vitulano. A generalized model for scratch detection. *IEEE Transactions on Image Processing*, 13(1):44–50, 2004. doi: 10.1109/TIP.2003.817231.

[4] Adrian Bulat, Jing Yang, and Georgios Tzimiropoulos. To learn image super-resolution, use a gan to learn how to do image degradation first. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[5] Harold C Burger, Christian J Schuler, and Stefan Harmeling. Image denoising: Can plain neural networks compete with bm3d? In *2012 IEEE conference on computer vision and pattern recognition*, pages 2392–2399. IEEE, 2012.

[6] 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 (CVPR)*, July 2017.

[7] 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 *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[8] Rong-Chi Chang, Yun-Long Sie, Su-Mei Chou, and T.K. Shih. Photo defect detection for image inpainting. In *Seventh IEEE International Symposium on Multimedia (ISM'05)*, pages 5 pp.–, 2005. doi: 10.1109/ISM.2005.91.

[9] Chang Chen, Zhiwei Xiong, Xinmei Tian, Zheng-Jun Zha, and Feng Wu. Camera lens super-resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[10] Michele Claus and Jan van Gemert. Videnn: Deep blind video denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[11] Victor Cornillere, Abdelaziz Djelouah, Wang Yifan, Olga Sorkine-Hornung, and Christopher Schroers. Blind image super-resolution with spatially variant degradations. *ACM Transactions on Graphics (TOG)*, 38(6):1–13, 2019.

[12] Xin Deng, Hao Wang, Mai Xu, Yichen Guo, Yuhang Song, and Li Yang. Lau-net: Latitude adaptive upscaling network for omnidirectional image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9189–9198, June 2021.

[13] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *European conference on computer vision*, pages 184–199. Springer, 2014.

[14] I. Giakoumis, N. Nikolaidis, and I. Pitas. Digital image processing techniques for the detection and removal of cracks in digitized paintings. *IEEE Transactions on Image Processing*, 15(1):178–188, 2006. doi: 10.1109/TIP.2005.860311.

[15] Jinjin Gu, Hannan Lu, Wangmeng Zuo, and Chao Dong. Blind super-resolution with iterative kernel correction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

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

[17] Jingwen He, Chao Dong, and Yu Qiao. Interactive multi-dimension modulation with dynamic controllable residual learning for image restoration. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 53–68. Springer, 2020.

[18] 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.

[19] Xuecai Hu, Haoyuan Mu, Xiangyu Zhang, Zilei Wang, Tieniu Tan, and Jian Sun. Meta-sr: A magnification-arbitrary network for super-resolution. 2019.

[20] Zhe Hu and Ming-Hsuan Yang. Learning good regions to deblur images. *International Journal of Computer Vision*, 115(3):345–362, 2015. doi: 10.1007/s11263-015-0821-1.

[21] Viren Jain and Sebastian Seung. Natural image denoising with convolutional networks. *Advances in neural information processing systems*, 21, 2008.

[22] Younghyun Jo, Seoung Wug Oh, Jaeyeon Kang, and Seon Joo Kim. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[23] Tarun Kalluri, Deepak Pathak, Manmohan Chandraker, and Du Tran. Flavr: Flow-agnostic video representations for fast frame interpolation. *arXiv preprint arXiv:2012.08512*, 2020.

[24] 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. doi: 10.1109/TCI.2016.2532323.

[25] Heewon Kim, Sungyong Baik, Myungsub Choi, Janghoon Choi, and Kyoung Mu Lee. Searching for controllable image restoration networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14234–14243, 2021.

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

[27] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[28] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[29] Juncheng Li, Faming Fang, Kangfu Mei, and Guixu Zhang. Multi-scale residual network for image super-resolution. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[30] Zhen Li, Jinglei Yang, Zheng Liu, Xiaomin Yang, Gwanggil Jeon, and Wei Wu. Feedback network for image super-resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[31] Ding Liu, Zhaowen Wang, Yuchen Fan, Xianming Liu, Zhangyang Wang, Shiyu Chang, and Thomas Huang. Robust video super-resolution with learned temporal dynamics. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[32] Matteo Maggioni, Yibin Huang, Cheng Li, Shuai Xiao, Zhongqian Fu, and Fenglong Song. Efficient multi-stage video denoising with recurrent spatio-temporal fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3466–3475, June 2021.

[33] Yiqun Mei, Yuchen Fan, Yuqian Zhou, Lichao Huang, Thomas S. Huang, and Honghui Shi. Image super-resolution with cross-scale non-local attention and exhaustive self-exemplars mining. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[34] Tomer Michaeli and Michal Irani. Nonparametric blind super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 945–952, 2013.

[35] Jinshan Pan, Haoran Bai, Jiangxin Dong, Jiawei Zhang, and Jinhui Tang. Deep blind video super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4811–4820, October 2021.

[36] Jinshan Pan, Haoran Bai, Jiangxin Dong, Jiawei Zhang, and Jinhui Tang. Deep blind video super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4811–4820, 2021.

[37] Seong-Jin Park, Hyeongseok Son, Sunghyun Cho, Ki-Sang Hong, and Seungyong Lee. Srfeat: Single image super-resolution with feature discrimination. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[38] Yajun Qiu, Ruxin Wang, Dapeng Tao, and Jun Cheng. Embedded block residual network: A recursive restoration model for single-image super-resolution. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[39] Dev Yashpal Sheth, Sreyas Mohan, Joshua Vincent, Ramon Manzorro, Peter A. Crozier, Mitesh M. Khapra, Eero P. Simoncelli, and Carlos Fernandez-Granda. Unsupervised deep video denoising. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021.

[40] Assaf Shocher, Nadav Cohen, and Michal Irani. "zero-shot" super-resolution using deep internal learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3118–3126, 2018.

[41] F. Stanco, Giovanni (Gianni) Ramponi, and Andrea Polo. Towards the automated restoration of old photographic prints: A survey. pages 370 – 374 vol.2, 10 2003. ISBN 0-7803-7763-X. doi: 10.1109/EURCON.2003.1248221.

[42] M. Suganuma, X. Liu, and T. Okatani. Attention-based adaptive selection of operations for image restoration in the presence of unknown combined distortions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[43] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[44] 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 (ICCV)*, Oct 2017.

[45] Matias Tassano, Julie Delon, and Thomas Veit. Dvdnet: A fast network for deep video denoising. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 1805–1809. IEEE, 2019.

[46] Matias Tassano, Julie Delon, and Thomas Veit. Fastdvdnet: Towards real-time deep video denoising without flow estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[47] 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.

[48] Ziyu Wan, Bo Zhang, Dongdong Chen, Pan Zhang, Dong Chen, Jing Liao, and Fang Wen. Bringing old photos back to life. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2747–2757, 2020.

[49] 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.

[50] 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.

[51] 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.

[52] Yifan Wang, Federico Perazzi, Brian McWilliams, Alexander Sorkine-Hornung, Olga Sorkine-Hornung, and Christopher Schroers. A fully progressive approach to single-image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[53] Junyuan Xie, Linli Xu, and Enhong Chen. Image denoising and inpainting with deep neural networks. In *Advances in neural information processing systems*, pages 341–349, 2012.

[54] Xiangyu Xu, Yongrui Ma, and Wenxiu Sun. Towards real scene super-resolution with raw images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[55] Yu-Syuan Xu, Shou-Yao Roy Tseng, Yu Tseng, Hsien-Kai Kuo, and Yi-Min Tsai. Unified dynamic convolutional network for super-resolution with variational degradations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

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

[57] Fuzhi Yang, Huan Yang, Jianlong Fu, Hongtao Lu, and Baining Guo. Learning texture transformer network for image super-resolution. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[58] Ke Yu, Chao Dong, Liang Lin, and Chen Change Loy. Crafting a toolchain for image restoration by deep reinforcement learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[59] Huanjing Yue, Cong Cao, Lei Liao, Ronghe Chu, and Jingyu Yang. Supervised raw video denoising with a benchmark dataset on dynamic scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[60] Kai Zhang, Luc Van Gool, and Radu Timofte. Deep unfolding network for image super-resolution. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[61] 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.

[62] Wenlong Zhang, Yihao Liu, Chao Dong, and Yu Qiao. Ranksrgan: Generative adversarial networks with ranker for image super-resolution. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[63] Xuaner Zhang, Qifeng Chen, Ren Ng, and Vladlen Koltun. Zoom to learn, learn to zoom. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[64] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[65] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[66] Yulun Zhang, Kai Li, Kunpeng Li, and Yun Fu. Mr image super-resolution with squeeze and excitation reasoning attention network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13425–13434, June 2021.