applied sciences

MDPI

Article

Video-Restoration-Net: Deep Generative Model with Non-Local Network for Inpainting and Super-Resolution Tasks

Yuanfeng Zheng, Yuchen Yan and Hao Jiang *

School of Electronic Information, Wuhan University, Wuhan 430072, China; zhengyuanfeng@zstp.edu.cn (Y.Z.); yyc@whu.edu.cn (Y.Y.)
* Correspondence: jh@whu.edu.cn

Abstract: Although deep learning-based approaches for video processing have been extensively investigated, the lack of generality in network construction makes it challenging for practical applications, particularly in video restoration. As a result, this paper presents a universal video restoration model that can simultaneously tackle video inpainting and super-resolution tasks. The network, called Video-Restoration-Net (VRN), consists of four components: (1) an encoder to extract features from each frame, (2) a non-local network that recombines features from adjacent frames or different locations of a given frame, (3) a decoder to restore the coarse video from the output of a non-local block, and (4) a refinement network to refine the coarse video on the frame level. The framework is trained in a three-step pipeline to improve training stability for both tasks. Specifically, we first suggest an automated technique to generate full video datasets for super-resolution reconstruction and another complete-incomplete video dataset for inpainting, respectively. A VRN is then trained to inpaint the incomplete videos. Meanwhile, the full video datasets are adopted to train another VRN frame-wisely and validate it against authoritative datasets. We show quantitative comparisons with several baseline models, achieving 40.5042 dB/0.99473 on PSNR/SSIM in the inpainting task, while during the SR task we obtained 28.41 dB/0.7953 and 27.25/0.8152 on BSD100 and Urban100, respectively. The qualitative comparisons demonstrate that our proposed model is able to complete masked regions and implement super-resolution reconstruction in videos of high quality. Furthermore, the above results show that our method has greater versatility both in video inpainting and super-resolution tasks compared to recent models.

Keywords: video restoration; video inpainting; super-resolution; non-local

Citation: Zheng, Y.; Yan, Y.; Jiang, H. Video-Restoration-Net: Deep Generative Model with Non-Local Network for Inpainting and Super-Resolution Tasks. *Appl. Sci.* **2023**, *13*, 10001. https://doi.org/10.3390/app131810001

Academic Editor: Pavel Lyakhov

Received: 1 June 2023
Revised: 27 August 2023
Accepted: 1 September 2023
Published: 5 September 2023

1. Introduction

Although videos are widely used in social life, they remain incomplete and indistinct in many situations. Therefore, video restoration is significant in these cases. However, no existing models can perform both video inpainting (also known as video completion or video hole filling) and super resolution (SR) synchronously. In this paper, we aim to reconstruct the missing region and increase the resolution in frames with a generalized model, leading to coherent and high-resolution results spatially in each frame and temporally along different frames.

In the situation of inpainting, pixels in the video frame may be corrupted during transmission, or some undesirable situations, such as watermarks or objects, need to be removed. Image inpainting has been extensively studied in the literature, which is consistent with video inpainting tasks. Texture synthesis methods in early works [1,2] match and copy background patches to fill the hole. However, they cannot handle the challenging cases where inpainting regions are complex and non-stationary. The major obstacle is that these methods cannot learn to understand the semantics of images and rely heavily on matching low-level, human-crafted features. Recently, deep generative inpainting methods based on convolutional neural networks (CNN) [3] and generative

adversarial networks (GANs) [4,5] have demonstrated great improvements over traditional methods. These methods learn to understand the semantics of images by training on large datasets and perform much better on non-stationary cases, such as incomplete objects, faces, complex scenes and many others. However, these methods for image inpainting have inferior performance when applied to the task of video inpainting, where the missing content in frames is required to be spatially coherent (in each frame) and temporally consistent (across frames). Although some work has attempted to extend image inpainting methods to videos, their methods perform poorly when both the background and holes are not static [6]. As a result, it is essential to combine spatio-temporal information not only from the current frame but also from the nearby frames.

The existing video inpainting algorithms can be divided into object-based and patch-based methods. In the object-based method, a video is divided into its background and foreground, which are inpainted separately and merged at the end of the algorithm [7,8]. However, these algorithms are not robust and produce artifacts when objects are in motion. On the other hand, patch-based methods search for suitable patches from global and local frames based on both 2D patches [9] and 3D patches [10–12]. Although patch-based methods have some successful cases, they still have some drawbacks, such as heavy computation and unstable performance in videos with motion and occlusion.

In the situation of super-resolution reconstruction, a non-linear mapping is used to reconstruct high-resolution (HR) images from low-resolution (LR) images. Although several methods have achieved good performance, they still remain intractable for the following ill-posed problem: how to model the mapping relationship from LR to HR images. Especially, an LR image with identical representation can be obtained from an infinite number of HR images by downscaling, resulting in a large space of possible functions during mapping. It is particularly difficult to learn an appropriate function in such a large space, which ultimately limits the capability of the model. Although several recent models have significantly improved SR performance, such as SRGAN [13], SRNTT [14], and RDN [15], they still suffer from the fact that the space of the possible mapping functions remains large.

Furthermore, the direct application of the above methods to video super resolution is not appropriate due to the particularities of videos. Previous research [16–18] on video SR still represents a simple extension of image super-resolution methods. Recent research [19–22] addresses the above challenges with more efficient methods consisting of alignment and fusion. However, these methods still suffer from poor performance when there is a lot of motion or severe blur in the video.

Based on the above discussions, there are still significant challenges in the existing techniques of video inpainting and video super resolution. Moreover, there is a significant disparity in the design of algorithmic models in these two domains, leading to the lack of a universal model. As a result, we introduce a generalized Video-Restoration-Net (VRN), supported by a non-local operation, to perform both tasks successively. Firstly, due to the lack of suitable datasets, we propose an automatic algorithm to synthesize full video datasets for super-resolution reconstruction, while another incomplete-complete dataset is constructed using large and varied hole-mask sequences simulating different real-world use cases. Specifically, some frames of the incomplete-complete datasets are shown in Figure 1. From top to bottom, we show examples of the collected videos, masks, and a mask sequence that we generated from a single mask. Based on the two synthetic datasets, we design a WGAN-style framework [23] to train a non-local deep generative network for inpainting and SR tasks. During inpainting tasks, we use an encoder–decoder model in the generator to extract the feature of each frame in the video and to restore frames based on the predicted feature, respectively. Between the encoder and decoder, we stack the feature maps of the frames and introduce a non-local network module [24] that models long-range, spatio-temporal dependencies to fill in the missing part of the inpainting task, while the frames should be input in an unstacked situation to recombine the non-local information from different locations of a given frame in the SR task. The recombined feature maps are

then unstacked and inputted frame-by-frame to the decoder to generate a coarse video. After the encoder–nonlocal-decoder construction, we design a network to refine each frame separately. On the other hand, we employ both global and local discriminators for frame inpainting [25]. Meanwhile, we perform the video SR task by adopting another VRN based on the full video datasets frame-wisely, assisting with a similar training course adopted in the video inpainting task. Both $\ell 1$ reconstruction loss and adversarial loss are used in supervision during training.

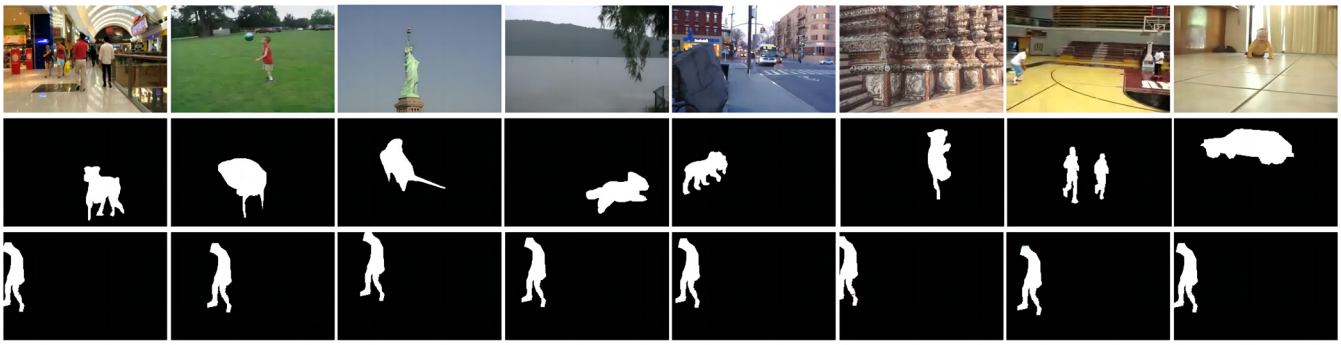

Figure 1. The datasets we synthesize for video inpainting.

Overall, we first present experiments on the complete-incomplete datasets we synthesize to demonstrate the effectiveness of our proposed method in video inpainting. Moreover, we adopt the full video datasets we synthesize as training datasets for SR reconstruction to explore the performance of our proposed method and validate it on public datasets, proving the universality of our Video-Restoration-Net. Our contributions can be summarized as follows:

- We present a deep generative non-local network for the task of video inpainting and frame-by-frame video SR reconstruction as an effective and generalized baseline. The proposed model is able to deal with various shapes of masks with divergent motion while reconstructing the generated low-resolution videos frame by frame.
- We propose an automatic algorithm to synthesize incomplete videos from complete ones, motivated by real-use cases of video inpainting.
- We present experiments through inpainting and SR results supported by ablation studies compared with several deep-learning-based baselines and show that our method based on a non-local module produces higher quality results in both video inpainting and frame-by-frame video SR tasks.

2. Related Work

2.1. Inpainting Methods

DL image inpainting algorithms have shown better performance than the traditional methods, and most of them are based on GANs. In [3], an encoder–decoder network is presented to restore the missing parts and an adversarial loss is employed to improve the quality of the inpainted image. In [25], a global–local discriminator is proposed that includes two networks to evaluate the consistency of the whole inpainted image and the quality of the restored missing part, respectively. Moreover, Iizuka et al. [25] replaced some channel-wise fully connected layers with dilated convolutions to increase the receptive fields of output neurons [26], also using a global–local discriminator and proposed a contextual attention layer that searches for matching background to reconstruct the foreground. Similarly, Song et al. [27] designed a patch-swap unit to match neural patches from the boundary to the hole. In Shift-Net [28], a special shift connection layer is added to the U-Net architecture [29] to fill in the missing regions of any shape with sharp structures and fine-detailed textures. Inspired by these image inpainting algorithms, we also adopt the

encoder–decoder-based architecture and the global–local discriminator design in our video inpainting model.

Yang et al. [30] propose a multi-task framework that incorporates structural constraints to generate clear edges. Liu et al. [31] introduce a mutual encoder–decoder network that jointly learns the CNN features that correspond to the same task. Structures and textures with various layers present a challenging task to model, particularly in creating a shared architecture that effectively complements both features. Wan et al. [32] adopt bidirectional attention and auto-regressive transformers to perform image inpainting. Although they promote diversity, their completion and inference performances are limited.

The video inpainting task was first proposed in 1998 [33], and there is still no DL solution. The author of [8] divides the video into foreground and background objects and iteratively paints the hole, frame by frame. The main disadvantages of these algorithms are their long computation times and artifacts. However, patch-based methods have demonstrated better performance. Newson et al. [11] find non-local patches containing occluded pixels through an approximate nearest neighbor search and iteratively optimize the global patch-based function that searches for patches and restores the hole. However, the algorithm proposed in [11] is very slow. Recently, a modified method of [11] has been proposed [12]. It uses optical flow, which has been used in some video inpainting work [10], to preserve temporal coherence and accelerate the algorithm. As a result, it achieves better results not only in terms of speed but also in terms of inpainting quality. Recently, the authors of [34] adopted an attention-based method to pay more attention to contextual information. The authors of [35] utilize 3D convolution and attention, which often produce results with limited temporal coherence due to their restricted temporal receptive fields. However, the recent models lack universality while also having limited improvement in PSNR and SSIM index and little improvement in visual effect. Thus, our work is focused on introducing a universal model capable of restoring both images and videos. Since [11] proposes a canonical method in terms of versatility, we select this as the baseline model for the inpainting task.

2.2. SR Methods

Since the seminal research of SRCNN [36], several explorations have been conducted to enhance the efficiency of image SR, such as interpolation-based approaches [37] and reconstruction-based methods [13–15,38,39]. In [13], an SRGAN based on GAN is proposed, which is the first model capable of deriving photorealistic images for 4× upscaling factors. In [38], a back-projection network (DBPN) supported by up-sampling and down-sampling layers is proposed for image SR reconstruction. Zhang et al. [39] built a deep architecture called RCAN supported by the channel attention mechanism. In [14], a typical RefSR algorithm named SRNTT is proposed to complete the local texture matching within the feature space and further add the matched textures to the output of the model. In [15], Zhang et al. propose a residual dense network (RDN), which takes full advantage of the hierarchical features acquired from the low-resolution image supported by the residual dense block. However, all of these still suffer from the fact that the space of possible mapping functions remains large, resulting in difficulty modeling a superior framework.

For video SR, temporal features show significant effects and have been extensively explored. In [19,21], they use optical flow to evaluate the motion between frames and perform warping. However, it is difficult to acquire accurate flow under large motions. In [20], the proposed TOFlow demonstrates that the standard optical flow cannot represent first-rank motion in the video SR task. Moreover, the advancement of DL methods greatly promotes this property in video deblurring [40,41]. In [41,42], they combine sequential frames without precise temporal alignment due to the fact that the appearance of blur seriously affects motion estimation. Different from these approaches, we aim to acquire non-local information from each frame due to the fact that the reconstruction of a pixel in a particular frame may rely on global features that do not exist in the receptive field of the convolution kernel.

In the recent research, an attention mechanism has been utilized in video restoration along with CNN. Isobe et al. [43] have categorized frames into various groups and created a temporal group attention module. Suin et al. [44] have suggested a reinforcement approach, supported by a learning-based framework with factorized spatio-temporal information. Furthermore, Cao et al. [45] recommend using self-attention within local patches of a video. Although these methods have significantly enhanced the performance of video SR, there remains a dearth of model universality. Our work pays more attention to constructing a generalized model that deals with video inpainting and frame-by-frame SR simultaneously, which is an urgent problem in the field of deep learning as no previous studies have succeeded in this field.

2.3. Non-Local Neural Network

Non-local neural networks [24], inspired by the classical non-local means method for image denoising, allow the network to capture long-range dependencies. The non-local operation in neural networks can be formulated as:

$$y_i = \frac{1}{C(x)} \sum_{\forall j} f(x_i, x_j) g(x_j) \tag{1}$$

Here, the output y at the position i is calculated from the value of the input x_i at the same position and the possible different positions of the input x_j, which have contributions to the position i. In this way, the respective field is enlarged to the whole video rather than being restricted to the local area of the convolution kernel as in the convolution operation. The architecture of the non-local block is shown in Figure 2. It takes one video as the input; T represents the number of frames, which is 32, and each frame contains 256 feature maps of size 20 × 30.

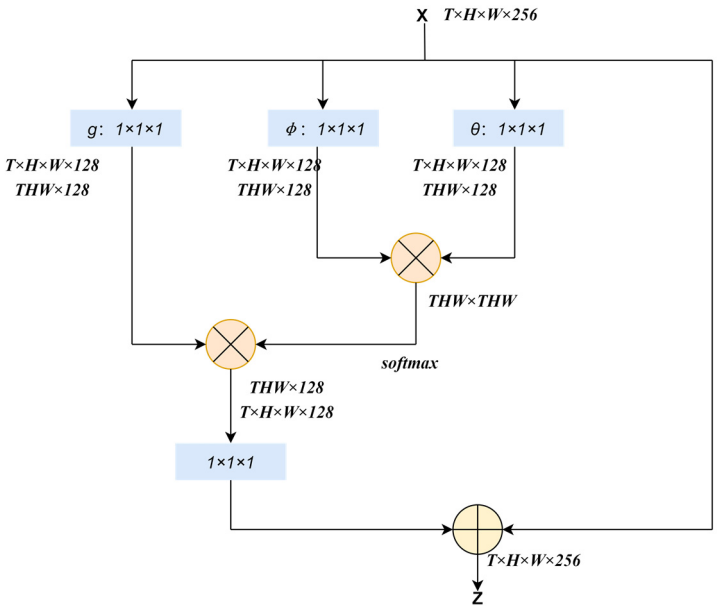

Figure 2. The non-local block.

Moreover, a modified, non-local recurrent network has been proposed for image restoration [46]. In this paper, we introduce a 3D non-local block, as shown in Figure 2, to the video inpainting task, which combines the spatio-temporal features of each frame, while adopting another 2D block to the SR task, which pays attention to different locations of a particular frame.

3. Materials and Methods

In this section, we introduce our data generation method and the proposed algorithm for the video inpainting and SR tasks.

To generate the data for two tasks, we collect some suitable videos from several existing datasets. We also select several appropriate masks for video inpainting. The details of the method are described in Section 3.1.

Our model is a WGAN-style [47] neural network. During the video inpainting task, we adopt an incomplete video, V_i, and its mask sequence, M, as inputs, the generator, G, outputs a restored coarse video as an intermediate output, V_m, and then a network refines the coarse video and outputs the final result, V_r, which should be as consistent as possible with the complete video, V_c. Similar to the image inpainting methods [25,27], two global–local discriminative networks are trained with G in an adversarial manner to help G produce more qualified results. One discriminates the quality of the restored missing regions and the other one evaluates the consistency of the restored missing parts in each frame. Our model for video inpainting is shown in Figure 3, and the details are shown in Tables 1–4. It takes an incomplete video and its mask sequence as input. The encoder–nonlocal-decoder part outputs a coarse inpainted video. The refinement network first restores only the missing region and then reconstructs the whole inpainted video. The input is also concatenated to the coarse video and the restored missing region.

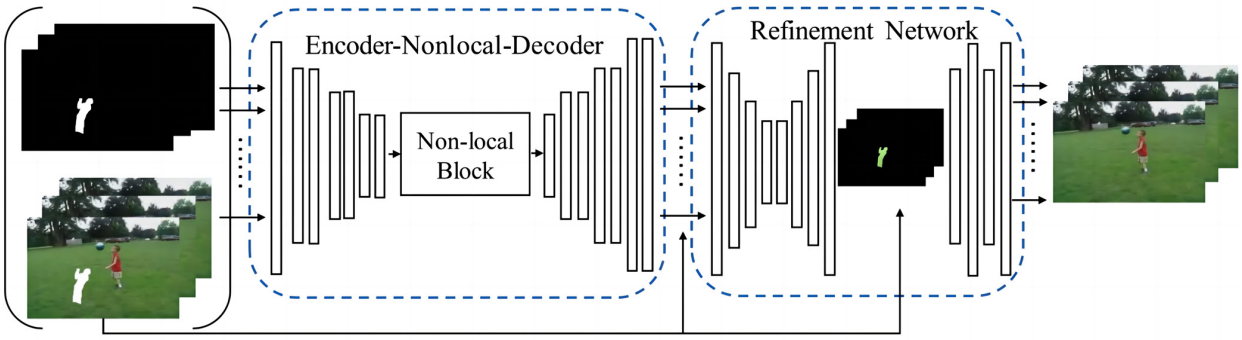

Figure 3. The proposed video for video inpainting.

Table 1. The details of encoder architecture.

Encoder	Type	Kernel	Stride	Channel
1	Input	-	-	6
2	Conv	3	1	32
3	Conv	3	2	64
4	Conv	3	1	64
5	Conv	3	2	128
6	Conv	3	1	128
7	Conv	3	2	256
8	Conv	3	1	256

Table 2. The details of decoder architecture.

Decoder	Type	Kernel	Stride	Channel
1	Conv	3	1	256
2	RConv	3	2	128
3	Conv	3	1	128
4	RConv	3	2	64
5	Conv	3	1	64
6	RConv	3	2	32
7	Conv	3	2	32
8	Conv	3	1	3

Table 3. The details of refinement network architecture.

R	Type	Connected with	Kernel	Stride	Channel
1	Input	Encoder-1	-	-	9
2	Conv	-	3	2	32
3	Conv	-	3	2	64
4	Conv	-	3	2	128
5	Conv	-	3	2	256
6	Conv	5	3	1	256
7	RConv	4	3	2	128
8	RConv	3	3	2	64
9	RConv	2	3	2	32
10	Conv	Encoder-1	3	1	3
11	Conv	-	3	2	32
12	RConv	-	3	2	32
13	Conv	-	3	2	32
14	RConv	-	3	2	32
15	Conv	-	3	1	3

Table 4. The details of discriminator network architecture.

D	Type	Kernel	Stride	Channel	BN
1	Input	-	-	3	-
2	Conv	3	1	16	N
3	Conv	3	2	16	Y
4	Conv	3	1	32	Y
5	Conv	3	2	32	Y
6	Conv	3	1	64	Y
7	Conv	3	2	64	Y
8	Conv	3	1	128	Y
9	Conv	3	2	128	Y
10	Flatten	-	-	614,400	-
11	Linear	-	-	1024	-

During the SR task, we adopt low-resolution videos generated by downsampling with a scaling factor of $\times 4$ and conduct the input frame-wisely. We utilize a 2D non-local block instead of the 3D block adopted in inpainting in order to reduce parameters and improve computational efficiency. Moreover, we use a global discriminator to evaluate the consistency of reconstructed high-resolution images, while other operations and model architecture are kept consistent with the video inpainting task, as shown in Figure 3.

3.1. Data Acquisition for Video Inpainting and SR

The performance of the neural network relies on reliable datasets. However, the property of the model is limited by the lack of datasets for video inpainting. To solve this problem, we need to build artificial video datasets with good diversity. The videos in our datasets should be in multiple scenes to ensure that the network will not be overfitted. In addition, the videos for the inpainting task should have the following characteristics: the masks should be different in size and shape to ensure that the network is able to handle a variety of masks rather than just one type. Secondly, the masks should be moving rather than static, and the moving track should be random.

Following the above principles, we collect 340 suitable videos for training from three datasets, UCF101 [48], MOT16 [49], and FCVID [50]. The datasets we build are rich in diversity, containing outdoor and indoor videos, sports and landscape videos, videos with and without camera motion, and with other difference characteristic. From these videos, we extract 937 clips, and each clip contains 32 frames with a 160×240 resolution as complete videos.

During the training course of the video SR task, the complete videos acquired above can be directly adopted as ground truth datasets, while they are treated as low-resolution input by conducting downsampling. Moreover, we adopt BSD100 [51] and Urban100 [50] as the test datasets for the SR task.

For the video inpainting task, we need to simulate the various shapes of the holes in the video by selecting some suitable video segmentation labels from the A2D datasets and moving them randomly to synthesize the hole mask sequence. As a result, the hole mask sequences are stochastic, regardless of shape, size, or motion tendency. Furthermore, in some sequences, there are even two or more holes. We synthesize the incomplete video using the video clips and the mask sequences; the pixels in the video are constrained between 0 and 1 and the pixels in the hole are set to 1. Finally, we generate 937 videos, of which 900 are for training and the rest are for testing. Some examples of collected masks, synthetic mask sequences, and videos are shown in Figure 1. Based on these datasets, it is possible to train a neural network using a supervised method and to quantificationally compare the inpainting results with other algorithms.

3.2. Generator Architecture Design

The proposed generator consists of four parts: an encoder, a non-local block, a decoder, and a refinement network. Since the encoder–decoder architecture has been used in the image algorithms [25,27], we also adopt this model in our work. The encoder, which contains seven convolutional layers, is used to extract the features of each frame of the video. The strides of the second, fourth, and sixth layer of the encoder are set to 2 for downsampling. The kernel sizes of the encoder are set to 3, and Elu [52] is adopted as the activation function. Moreover, the decoder, which has a corresponding structure with the encoder and uses resize-convolution for upsampling, is used to restore the inpainting and LR frames from the processed features.

During the inpainting task, we stack the feature map outputs of the encoder between the encoder and the decoder, and then the non-local block is used to combine the spatio-temporal features. Finally, we unstack the feature maps and feed them into the decoder to restore the inpainted frames, respectively. Due to the fact that a missing part of the current frame potentially could appear in other frames in the video, especially in the far frames where the missing part and low resolution case have coherence in spatio-temporal data, we need to consider a long dependency during the inpainting task. Therefore, the non-local neural network becomes very suitable for this task because the non-local block takes the feature maps of the whole video as input and computes the relationship between all positions. As shown in Figure 2, the feature maps are parallelly input into three $1 \times 1 \times 1$ 3D convolution layers, θ, ϕ and g, and then we can obtain an affinity representation of the relationship between each position, i, in the video and its corresponding position, j, through matrix multiplication and softmax operation. The convolution layer, g, outputs a representation of the input feature at the position j. These two results are added to the input feature maps to produce the final result after matrix multiplication and 3D convolution. In this way, the receptive field is extended to the entire video rather than the field of the convolution kernel. Thus, the information of all the other positions is considered not only in the current frame but also in the others. The details are shown in Tables 1 and 2.

Different from the encoder and decoder architecture adopted in the inpainting task, several modifications are implemented in the SR task. The main distinction is that we adopt 2D convolutional layers due to the fact that we conduct the input frame-wisely during the training course, while the other configuration of the network keeps consistent with that in the inpainting task.

The last part of the generator is a refining network, which is a dual-task network that takes the output of the decoder and the incomplete video as input during the inpainting task while taking the individual output of the decoder as input during the SR task. First, a U-net takes the ground truth frame as a label, reconstructing the inpainting and high-resolution frames, respectively. In this part, we continuously downsample and upsample

the input four times by using convolution layers with stride 2 and resize-convolution layers. Then, 5 convolution layers are applied to restore the final result. The details are shown in Table 3.

3.3. Discriminator Architecture Design

Our discriminator is built at the frame level; we evaluate one video at each step and treat the number of frames as the batch size. The architecture of the network is similar to that of SRGAN [13]. There are 8 convolutional layers in the architecture and half of them are with stride 2. We also use batch normalization and leaky ReLU in this architecture. The details of the discriminator are shown in Table 4.

In the video inpainting task, we use the above two networks as the global–local [25,27] discriminator during training, which is different from the SR task where only the global discriminator is used. Furthermore, the global–local discriminator is used in different training steps, which are introduced in Section 3.5. The local discriminator takes the restored missing region as input to evaluate its quality, and the global one takes the restored whole image as input to judge the consistency between the restored part and the complete part.

3.4. The WGAN Style Training Loss

Since the original GAN is always faced with the problem of disappearing gradients, the WGAN is proposed as a solution, which uses the Wasserstein distance to measure the gap between the real data distribution and the fake data distribution. The WGAN can be described as:

$$\min_{G} \max_{D} \mathbb{E}_{x\sim\mathbb{P}_r}[D(x)] - \mathbb{E}_{G(z)\sim\mathbb{P}_g}[D(G(z))] \tag{2}$$

where D and G represent the discriminator and generator, respectively. z is random noise in the WGAN, but it is the incomplete video with its mask sequence in our work.

As mentioned above, the same model is adopted in both video inpainting and SR tasks and trained in two separate courses. The training loss of the video inpainting and SR tasks can be described as follows.

During the video inpainting task, the discriminator we design contains two parts: the global one evaluates the consistency of the restored missing part with the entire frame, and the local one judges the quality of the restored missing part.

For the global discriminator, we put the restored missing region back into the incomplete video; it can be described as:

$$V_g = V_r \odot M + V_i \odot (1 - M) \tag{3}$$

where M is the mask sequence, whose pixels are 1 in the missing region and 0 in the rest of the video.

We use V_g as the input of the global discriminator, and the label for complete video is V_c in both tasks. So the loss of global discriminator D_g is

$$l_{D_g} = \mathbb{E}_{V_r\sim\mathbb{P}_g}[D_g(V_g)] - \mathbb{E}_{V_c\sim\mathbb{P}_r}[D_g(V_c)] \tag{4}$$

For the local WGAN, we only focus on the quality of the restored missing-part frame. Therefore, the loss of local discriminator, D_l, can be written as:

$$l_{D_l} = \mathbb{E}_{V_r\sim\mathbb{P}_g}[D_l(V_r \odot M)] - \mathbb{E}_{V_c\sim\mathbb{P}_r}[D_l(V_c \odot M)] \tag{5}$$

On the other hand, we combine the $L1$ loss with the loss of WGAN to train our generator. We tried 3 forms of L1 loss. The first one is

$$l_{L1} = \|V_r - V_c\| \odot M \tag{6}$$

where we only want to compute the gap in the mask. However, this loss is not able to guide the network in learning how to restore the video. The output is chaotic and the loss does not decrease during training.

Whereafter, we realize that it is necessary to compute the loss of the full video, so we simply compute the Manhattan distance between the inpainting video and the ground truth, which can be described as:

$$l_{L1} = \|V_r - V_c\| \tag{7}$$

Nevertheless, we finally find that increasing the loss of the hole region 10 times to increase the attention to the area is helpful in training the network. Consequently, the $L1$ loss between the restored video, V_r, and the ground truth (GT), V_c, can be written as:

$$l_{L1} = 10\|V_r - V_c\| \odot M + \|V_r - V_c\| \odot (1 - M) \tag{8}$$

and the loss of generator in the video inpainting task is:

$$l_G = l_{L1} - 10^{-3} \times (\mathbb{E}_{V_r \sim \mathbb{P}_g}[D_g(V_g)] + \mathbb{E}_{V_r \sim \mathbb{P}_g}[D_l(V_r \odot M)]) \tag{9}$$

Intuitively, we use the $L1$ loss to directly regress the missing region to the complete video, and at the same time, WGAN trains the generator with adversarial gradients to make the result much more real.

During the video SR task, the discriminator we design contains only the global one, different from the inpainting task, which evaluates the consistency of the reconstructed high-resolution part with the entire frame.

In detail, we directly adopt the generated high-resolution frame, I_{HR}, as the input of the global discriminator and the label for the ground truth is I_{GT}. The loss of the global discriminator, $D_{g'}$, is

$$l_{D_{g'}} = \mathbb{E}_{I_{HR} \sim \mathbb{P}_g}[D_{g'}(I_{HR})] - \mathbb{E}_{I_{GT} \sim \mathbb{P}_r}[D_{g'}(I_{GT})] \tag{10}$$

Furthermore, we calculate the Manhattan distance between the generated high-resolution frame and ground truth with Equation (7) and obtain the loss of generator in the video SR task, formulated as:

$$l_{G'} = \|I_{HR} - I_{GT}\| - 10^{-3} \times (\mathbb{E}_{I_{HR} \sim \mathbb{P}_g}[D_{g'}(I_{HR})] + \mathbb{E}_{I_{GT} \sim \mathbb{P}_r}[D_{g'}(I_{GT})]) \tag{11}$$

3.5. Training Methods

Although we set a series of loss functions to guide the training, it is still not easy to train our model directly due to the deep architecture of our model and the complexity of the video inpainting and SR tasks. As a result, we train our model in two separate situations, each with 3 identical steps.

In the first step, we train the encoder–decoder with Mean Square Error loss for 100 epochs. The network takes each frame in our datasets as input and restores it. As a result, the encoder is able to extract reliable features for restoration, and the decoder can restore the frame based on these features.

Subsequently, we add the non-local block between the encoder and decoder and train them for 500 epochs. We adopt the incomplete video and its mask sequence as input for the video inpainting task and the low-resolution video as input for the SR task and output a corresponding restored and high-resolution video. We use the global–local discriminator in the inpainting task and the global discriminator in SR task, respectively, to judge the gap between the output video and the ground truth. We mainly desire to train the non-local block to learn to recombine the features from incomplete to complete. In this step, the parameters of the non-local decoder are optimized.

Finally, we train the whole encoder–nonlocal-decoder refinement model for 500 epochs. In the first 300 epochs of this step, we only update the parameters of the refined network and then the whole model is updated.

Since WAGN cannot be trained by momentum-based algorithms such as the Adam optimizer, we train our model on the RMSProp optimizer with a learning rate of 0.0001.

4. Results and Discussions

Our model is implemented on TensorFlow v1.13.1, CUDNN v7.4, and CUDA v10.0, and it runs on hardware with a GPU Nvidia GTX2080.

In the video inpainting task, we train the proposed model on 900 videos and evaluate it on 37 videos. Due to the face that the non-local block occupies a large amount of GPU memory, we restrict the video to 32 frames with a size of 160×240. It takes more than ten hours per epoch. In the SR task, we adopt the full video datasets for training and further evaluate our model on BSD100 and Urban100.

4.1. Quantitative Comparisons

During the inpainting task, as there were no datasets for video inpainting before, the evaluation is always based on qualitative comparisons. Although the inpainting result is not unique, which means that every result is acceptable as long as it is natural and realistic, the gap between the inpainting result, and the ground truth can still be used as a standard to evaluate the quality of the algorithms.

To evaluate our algorithm quantitatively, we compare our model with a typical patch-based algorithm [11]. This approach is considered canonical in terms of versatility, enabling us to confirm the versatility of our algorithm, while recent models are incapable of handling both inpainting and SR tasks. In addition, the code for [11] is publicly available online, so we can compare it with our synthetic test data.

However, the recent models lack universality, have limited improvement in PSNR and SSIM index and have little improvement in visual effect. Thus, our work is focused on introducing a universal model capable of restoring both images and videos. Since the method used in [11] is a canonical method in terms of versatility, we select it as the baseline model for the inpainting task.

We employ five assessment criteria, namely peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), mean L1 error, mean L2 error, and perceptual loss. The L1 and L2 errors intuitively reflect the gap between two images at the pixel level. PSNR is based on error sensitivity, and SSIM evaluates the image in terms of brightness, contrast, and structure. Perceptual loss, which is widely used in super-resolution tasks, is a function that evaluates the similarity between two images based on high-level features. Here, we use the feature of the second full connection layer of the VGG16 model as the standard. For L1 error, L2 error, and perceptual loss, a smaller value means better image quality. However, the opposite is true for PSNR and SSIM.

The average results of the five indexes on the synthetic test datasets are shown in Table 5. Our model performs better on PSNR, SSIM, L1 loss, and L2 loss, while the performance on perceptual loss is similar. This is mainly due to the non-local architecture, which considers mostly the spatio-temporal features in each frame. Moreover, as the recent models [47–51] have limited improvement in PSNR and SSIM index and little improvement in visual effect, the results of our model can already meet the application requirements of most scenarios, and our approach has greater universality.

Table 5. The comparison of inpainting results between [11] and our model.

Methods	PSNR	SSIM	L1	L2	VGG
[11]	40.2829	0.9917	0.6276	3.6390	0.1750
Ours	40.5042	0.99473	0.5484	2.9696	0.2411

Long computing times are an always-standing problem in video inpainting tasks. This is because a large amount of time is spent on patch seeking, matching, and any other options. For the deep learning algorithms, the parameters have been confirmed in training stage. Correspondingly, the calculation becomes extremely simple in the testing step. When inpainting a 32-frame video with a size of 160 × 240, the patch-based algorithm [11], when implemented on Matlab, takes about 6.5 min, but our model only consumes about 1.5 s on TensorFlow. Furthermore, the time of inpainting an incomplete sample frame cost about 0.05 s on our proposed model, while that of [11] costs about 12 s. Although the implement platform is different, the operation time is cut down greatly.

During the SR task, we conduct quantitative comparisons to demonstrate the advantages of our proposed model. All of the LR images are generated by bicubic downscaling with a scaling factor of ×4 from the HR images according to the standard protocol. We conduct the comparisons on benchmark datasets using the quality metrics of PSNR and SSIM. We compare our proposed model with several representative methods, including SRGAN [13], SRNTT [14], and RDN [15], as shown in Table 6. For the BSD100 and Urban100 datasets, our method yields excellent properties and outperforms the other methods on PSNR and SSIM. It reveals that the non-local architecture paid more attention to the different locations within a particular frame. Moreover, the above results also demonstrate the universality of our method in video inpainting and SR tasks, while no existing methods can deal with both situations.

Table 6. The comparison of SR results between SRGAN, SRNTT, RDN, and our model.

Datasets	Methods	PSNR	SSIM
BSD100	Bicubic	25.96	0.6675
	SRGAN	25.16	0.6688
	SRNTT	24.07	0.7290
	RDN	27.80	0.7434
	Ours	28.41	0.7953
Urban100	Bicubic	23.14	0.6577
	SRGAN	22.59	0.6780
	SRNTT	25.50	0.7830
	RDN	26.82	0.8069
	Ours	27.25	0.8152

4.2. Qualitative Comparisons

During the inpainting task, we show five inpainted frames of the test datasets in Figure 4. The top row is the ground-truth complete frame. The middle row is the result of the method used in [11], while the bottom row is the result of ours. Both [11]'s and our model can successfully inpaint the frame, as shown in (a). However, the method of [11] misses the fence in (d), so obviously our result is much closer to the ground truth, although their result is also acceptable.

However, the method proposed in [11] also has the problem that the inpainted hole region has an obvious boundary. We magnify the inpainted holes of (b), (c), and (e) in Figure 5 to compare the details. The red boxes in the image describe the parts that need to be inpainted. In Figure 5a, the boundary of the image of [11] is apparently disharmonious. Our result is much better but a little lacking in texture. In example (b), a profile of the missing region can be clearly observed in the result of [11], and our result has the same problem but looks brighter. As (c) shows, the result of [11] loses some details of the frame, but we restore them successfully. The above results further verify that the non-local mechanism is efficient through learning spatio-temporal features from each frame.

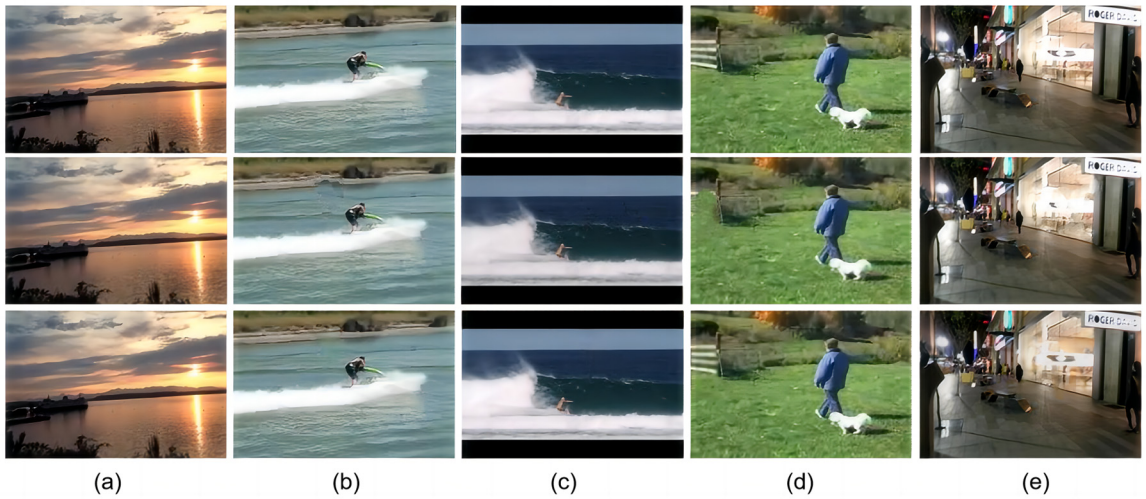

Figure 4. Comparative inpainting results on our synthetic test datasets between [11] and our model.

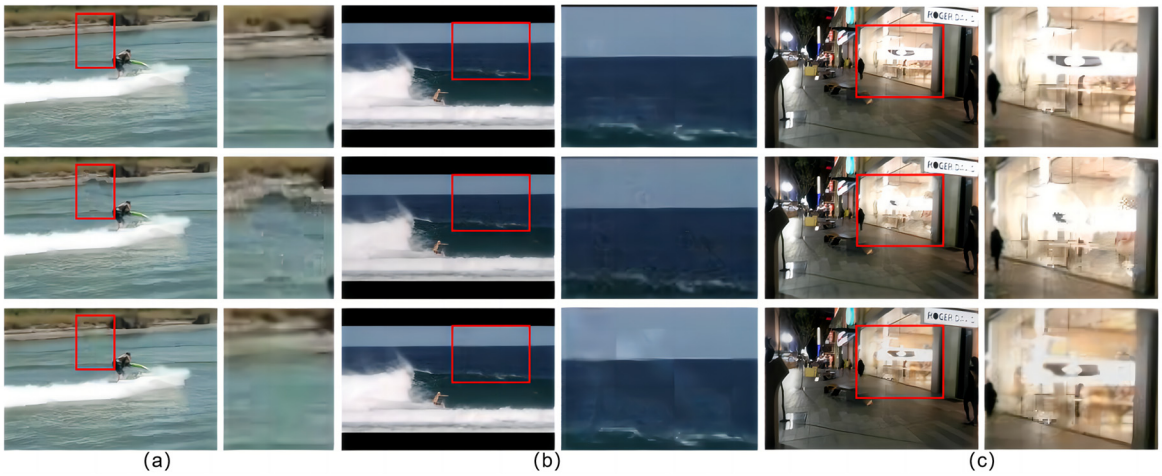

Figure 5. Comparison of the inpainted missing part between [11] and our model.

In [26], a texture pyramid and approximate nearest neighbor (ANN) search are used to extract patches to restore the hole, but this model is still not as reliable as ours. In our model, we first train the encoder–decoder to ensure that the encoder is able to extract reliable features for reconstruction and the non-local block is able to find useful information from non-local positions. As a result, under the reliable features and non-local information recombination, the following units will go in the right direction.

During the SR task, we also conduct qualitative analyses in terms of visual quality and texture enrichment, making comparisons between bicubic interpolation, ground truth (GT), SRGAN [13], SRNTT [14], RDN [15], and our model. The results of two images are shown in Figures 6 and 7 with a scaling factor of $\times 4$. The corresponding quality metrics of PSNR and SSIM and the mean square error (MSE) belonging to each SR image are also exhibited below. The exhibited images are from BSD100 and Urban100, respectively.

Figure 6. Visual results for the comparison on a BSD100 random image between bicubic interpolation, GT, SRGAN, SRNTT, RDN, and our model with scaling factor ×4.

Figure 7. Visual results of comparison on an Urban100 random image between bicubic interpolation, GT, SRGAN, SRNTT, RDN, and our model with scaling factor ×4.

As shown in Figures 6 and 7, the red box in the image describe the parts that need to be restored. The compared methods produce blurred edges and noticeable artifacts, while our model achieves higher visual quality by recovering and presenting sharper and clearer edges. In addition, the three quality metrics of two images achieved by our model also outperform the other methods. The reconstructed images generated by our model are more faithful to the GT images, demonstrating that the non-local architecture we adopt is much more efficient by combining non-local information.

All these qualitative comparisons of video inpainting and SR tasks further indicate the powerful versatility of our method in completing a variety of tasks, and the results are very competitive.

This is of great significance to the development of a general model in the field of video restoration.

4.3. Validity of Non-Local Block

To evaluate the validity of the non-local block in our model, we conduct a comparison between the non-local block and 3D convolution, which is a widely used strategy in video processing. This part is performed on an inpainting task to evaluate the global and local performance. Based on the trained encoder–decoder model, we replace the non-local block with a 3D convolution layer and train the encoder–3D-Conv–decoder model for 100 epochs. The loss of the generator during the 100 epochs is shown in Figure 8.

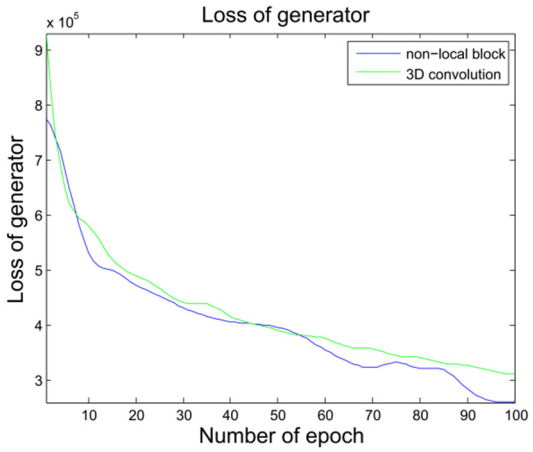

Figure 8. The comparison between the loss of generator with non-local block and 3Dconv.

The non-local block always gains a lower loss than the 3D convolution. The non-local block performs better as long as the non-local block has a larger respective field. Although the 3D convolution can combine the information from different frames, it is only able to combine the local features, which are limited by the size of the convolution kernel. However, the non-local block explores the relationship between different frames and positions at the pixel level by combining non-local spatio-temporal information. As a result, the non-local block has a stronger ability to find the useful information from distant frames and positions.

In addition, we are also inquisitive about how many non-local blocks are sufficient for this task. We adopt one, three, and five non-local blocks, respectively, in our encoder–nonlocal-decoder model and train the model adequately. The comparison result of the test datasets is shown in Table 7. We find that the performance becomes better when more blocks are used. Nevertheless, the calculation quantity also increases dramatically; the five-block version requires about 1.2 h for each epoch during training. The three-block version gains a similar performance with only 35 min. Therefore, we take up the three-block version in our work.

Table 7. The comparison of inpainting results between [11] and our model. The evaluation on different number of non-local block.

Num	PSNR	SSIM	L1 Loss	L2 Loss
1	35.8305	0.9887	0.5601	3.8708
3	40.5256	0.9916	0.4682	2.9415
5	40.7648	0.9922	0.4355	2.6395

4.4. Validity of Refinement Network

During the research of the inpainting task, we find that the encoder–nonlocal-decoder model is only able to restore a coarse result, as shown in Figure 9 shows. The top row is the result of the decoder and the bottom row is the refined result. This is caused by the damage of the image texture in the deep network. To solve this problem, we design an additional refinement network. The U-net architecture is first used to restore the hole region only for refining the texture of this region. Subsequently, four convolution layers are adopted to fill

the refined region back into the video and ensure coherence. As shown in Figure 7, the red box in the image describe the parts that need to be restored. It can be easily found that the texture in the hole region is successfully restored after the refinement of the network.

Figure 9. The comparison between the output of decoder and the whole network.

5. Conclusions

In this paper, we propose a method to generate the data for the video inpainting and SR task, making it possible to train a neural network in a supervised way. We design a deep model and introduce non-local blocks to our work to combine the non-local features, further proposing a generalized model that efficiently deals with the inpainting and super-resolution reconstruction tasks, which is also the first generalized model. The proposed model is able to successfully inpaint an incomplete video and generate SR frames. The comprehensive benchmark evaluations demonstrate that our model performs superiorly over representative methods, revealing the powerful versatility of our method in video inpainting and SR tasks.

Although our model may not perform as well as a model that is dedicated to handling a single task, we may compromise some single-task performance due to our pursuit of universality. However, the recent models have limited improvement in PSNR and SSIM index and little improvement in visual effect. The results of our model can already meet the application requirements of most scenarios, while pioneering the field of research on model universality, which is of great significance. In the future, our work will primarily concentrate on two elements. Firstly, we will modify our proposed model to enhance performance on a single task. Secondly, we will continue to explore more efficient universal models for video restoration.

Author Contributions: Conceptualization, H.J.; methodology, Y.Z.; software, Y.Z.; validation, Y.Y.; formal analysis, Y.Z.; investigation, Y.Y.; resources, H.J.; data curation, Y.Z.; writing—original draft preparation, Y.Z. and Y.Y.; writing—review and editing, Y.Z. and Y.Y.; visualization, Y.Y.; supervision, Y.Y.; project administration, Y.Y.; funding acquisition, Y.Y. All authors have read and agreed to the published version of the manuscript.

Funding: Funding through National Natural Science Foundation of China Enterprise Innovation and Development Joint Fund Key Project (Grant No. U19B2004).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data included in this study are available upon request by contact with the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Barnes, C.; Shechtman, E.; Finkelstein, A.; Goldman, D.B. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.* **2009**, *28*, 24. [CrossRef]
2. Hays, J.; Efros, A.A. Scene completion using millions of photographs. *ACM Trans. Graph. ToG* **2007**, *26*, 4-es. [CrossRef]
3. Pathak, D.; Krahenbuhl, P.; Donahue, J.; Darrell, T.; Efros, A.A. Context encoders: Feature learning by inpainting. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2536–2544.
4. Yang, C.; Lu, X.; Lin, Z.; Shechtman, E.; Wang, O.; Li, H. High-resolution image inpainting using multi-scale neural patch synthesis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6721–6729.
5. Yu, J.; Lin, Z.; Yang, J.; Shen, X.; Lu, X.; Huang, T.S. Free-form image inpainting with gated convolution. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 4471–4480.
6. Liu, M.; Chen, S.; Liu, J.; Tang, X. Video completion via motion guided spatial-temporal global optimization. In Proceedings of the 17th ACM International Conference on Multimedia, Beijing, China, 19–24 October 2009; pp. 537–540.
7. Ling, C.H.; Lin, C.W.; Su, C.W.; Chen, Y.S.; Liao, H.Y.M. Virtual contour guided video object inpainting using posture mapping and retrieval. *IEEE Trans. Multimed.* **2010**, *13*, 292–302. [CrossRef]
8. Granados, M.; Kim, K.I.; Tompkin, J.; Kautz, J.; Theobalt, C. Background inpainting for videos with dynamic objects and a free-moving camera. In Proceedings of the Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; Proceedings, Part I 12. Springer: Berlin/Heidelberg, Germany, 2012; pp. 682–695.
9. Barnes, C.; Shechtman, E.; Goldman, D.B.; Finkelstein, A. The generalized patchmatch correspondence algorithm. In Proceedings of the Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, 5–11 September 2010; Proceedings, Part III 11. Springer: Berlin/Heidelberg, Germany, 2010; pp. 29–43.
10. Huang, J.B.; Kang, S.B.; Ahuja, N.; Kopf, J. Temporally coherent completion of dynamic video. *ACM Trans. Graph.* **2016**, *35*, 1–11. [CrossRef]
11. Newson, A.; Almansa, A.; Fradet, M.; Gousseau, Y.; Pérez, P. Video inpainting of complex scenes. *Siam J. Imaging Sci.* **2014**, *7*, 1993–2019. [CrossRef]
12. Le, T.T.; Almansa, A.; Gousseau, Y.; Masnou, S. Motion-consistent video inpainting. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 2094–2098.
13. Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.P.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4681–4690.
14. Zhang, Z.; Wang, Z.; Lin, Z.; Qi, H. Image super-resolution by neural texture transfer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7982–7991.
15. Zhang, Y.; Tian, Y.; Kong, Y.; Zhong, B.; Fu, Y. Residual dense network for image super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2472–2481.
16. Takeda, H.; Milanfar, P.; Protter, M.; Elad, M. Super-resolution without explicit subpixel motion estimation. *IEEE Trans. Image Process.* **2009**, *18*, 1958–1975. [CrossRef]
17. Dai, Q.; Yoo, S.; Kappeler, A.; Katsaggelos, A.K. Dictionary-based multiple frame video super-resolution. In Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, Canada, 27–30 September 2015; pp. 83–87.
18. Liao, R.; Tao, X.; Li, R.; Ma, Z.; Jia, J. Video super-resolution via deep draft-ensemble learning. In Proceedings of the IEEE International Conference on Computer Vision, Boston, MA, USA, 7–12 June 2015; pp. 531–539.
19. Caballero, J.; Ledig, C.; Aitken, A.; Acosta, A.; Totz, J.; Wang, Z.; Shi, W. Real-time video super-resolution with spatio-temporal networks and motion compensation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4778–4787.
20. Xue, T.; Chen, B.; Wu, J.; Wei, D.; Freeman, W.T. Video enhancement with task-oriented flow. *Int. J. Comput. Vis.* **2019**, *127*, 1106–1125. [CrossRef]
21. Tao, X.; Gao, H.; Liao, R.; Wang, J.; Jia, J. Detail-revealing deep video super-resolution. In Proceedings of the IEEE International Conference on Computer Vision, Honolulu, HI, USA, 21–26 July 2017; pp. 4472–4480.
22. Sajjadi, M.S.; Vemulapalli, R.; Brown, M. Frame-recurrent video super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6626–6634.
23. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144. [CrossRef]
24. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7794–7803.
25. Iizuka, S.; Simo-Serra, E.; Ishikawa, H. Globally and locally consistent image completion. *ACM Trans. Graph.* **2017**, *36*, 1–14. [CrossRef]
26. Yu, J.; Lin, Z.; Yang, J.; Shen, X.; Lu, X.; Huang, T.S. Generative image inpainting with contextual attention. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5505–5514.
27. Song, Y.; Yang, C.; Lin, Z.; Liu, X.; Huang, Q.; Li, H.; Kuo, C.C.J. Contextual-based image inpainting: Infer, match, and translate. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.

28. Yan, Z.; Li, X.; Li, M.; Zuo, W.; Shan, S. Shift-net: Image inpainting via deep feature rearrangement. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 1–17.
29. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015; Proceedings, Part III 18. Springer International Publishing: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
30. Yang, J.; Qi, Z.; Shi, Y. Learning to incorporate structure knowledge for image inpainting. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12605–12612, No. 07.
31. Liu, H.; Jiang, B.; Song, Y.; Huang, W.; Yang, C. Rethinking image inpainting via a mutual encoder-decoder with feature equalizations. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part II 16. Springer International Publishing: Berlin/Heidelberg, Germany; pp. 725–741.
32. Wan, Z.; Zhang, J.; Chen, D.; Liao, J. High-fidelity pluralistic image completion with transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Online Meeting, 11–17 October 2021; pp. 4692–4701.
33. Masnou, S.; Morel, J.M. Level lines based disocclusion. In Proceedings of the 1998 International Conference on Image Processing, ICIP98 (Cat. No. 98CB36269), Chicago, IL, USA, 4–7 October 1998; pp. 259–263.
34. Li, A.; Zhao, S.; Ma, X.; Gong, M.; Qi, J.; Zhang, R.; Tao, D.; Kotagiri, R. Short-term and long-term context aggregation network for video inpainting. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part IV 16. Springer International Publishing: Berlin/Heidelberg, Germany, 2020; pp. 728–743.
35. Wang, C.; Huang, H.; Han, X.; Wang, J. Video inpainting by jointly learning temporal structure and spatial details. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27–28 January 2019; Volume 33, No. 01. pp. 5232–5239.
36. Dong, C.; Loy, C.C.; He, K.; Tang, X. Learning a deep convolutional network for image super-resolution. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Proceedings, Part IV 13. Springer International Publishing: Berlin/Heidelberg, Germany, 2014; pp. 184–199.
37. Hou, H.; Andrews, H. Cubic splines for image interpolation and digital filtering. *IEEE Trans. Acoust. Speech Signal Process.* **1978**, *26*, 508–517.
38. Haris, M.; Shakhnarovich, G.; Ukita, N. Deep back-projection networks for super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1664–1673.
39. Zhang, Y.; Li, K.; Li, K.; Wang, L.; Zhong, B.; Fu, Y. Image super-resolution using very deep residual channel attention networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 286–301.
40. Pan, L.; Dai, Y.; Liu, M.; Porikli, F. Simultaneous stereo video deblurring and scene flow estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4382–4391.
41. Su, S.; Delbracio, M.; Wang, J.; Sapiro, G.; Heidrich, W.; Wang, O. Deep video deblurring for hand-held cameras. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1279–1288.
42. Zhang, K.; Luo, W.; Zhong, Y.; Ma, L.; Liu, W.; Li, H. Adversarial spatio-temporal learning for video deblurring. *IEEE Trans. Image Process.* **2018**, *28*, 291–301. [CrossRef] [PubMed]
43. Isobe, T.; Li, S.; Jia, X.; Yuan, S.; Slabaugh, G.; Xu, C.; Wang, S.; Tian, Q. Video super-resolution with temporal group attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 8008–8017.
44. Suin, M.; Rajagopalan, A.N. Gated spatio-temporal attention-guided video deblurring. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–21 June 2021; pp. 7802–7811.
45. Cao, J.; Li, Y.; Zhang, K.; Liang, J.; Van Gool, L. Video Super-Resolution Transformer. *arXiv* **2021**, arXiv:2106.06847.
46. Liu, D.; Wen, B.; Fan, Y.; Loy, C.C.; Huang, T.S. Non-local recurrent network for image restoration. *arXiv* **2018**, arXiv:1806.02919.
47. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein generative adversarial networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 214–223.
48. Soomro, K.; Zamir, A.R.; Shah, M. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv* **2012**, arXiv:1212.0402.
49. Milan, A.; Leal-Taixé, L.; Reid, I.; Roth, S.; Schindler, K. MOT16: A benchmark for multi-object tracking. *arXiv* **2016**, arXiv:1603.00831.
50. Jiang, Y.G.; Wu, Z.; Wang, J.; Xue, X.; Chang, S.F. Exploiting feature and class relationships in video categorization with regularized deep neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 352–364. [CrossRef]

51. Arbelaez, P.; Maire, M.; Fowlkes, C.; Malik, J. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *33*, 898–916. [CrossRef] [PubMed]
52. Clevert, D.A.; Unterthiner, T.; Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus). *arXiv* **2015**, arXiv:1511.07289.