Artificial Intelligence Review (2023) 56:5317–5364
https://doi.org/10.1007/s10462-022-10302-5

RESEARCH

Video restoration based on deep learning: a comprehensive survey

Claudio Rota[1] · **Marco Buzzelli**[1] · **Simone Bianco**[1] · **Raimondo Schettini**[1]

Published online: 27 October 2022

Abstract
Video restoration concerns the recovery of a clean video sequence starting from its degraded version. Different video restoration tasks exist, including denoising, deblurring, super-resolution, and reduction of compression artifacts. In this paper, we provide a comprehensive review of the main features of existing video restoration methods based on deep learning. We focus our attention on the main architectural components, strategies for motion handling, and loss functions. We analyze the standard benchmark datasets and use them to summarize the performance of video restoration methods, both in terms of effectiveness and efficiency. In conclusion, the main challenges and future research directions in video restoration using deep learning are highlighted.

Keywords Video restoration · Super-resolution · Denoising · Deblurring · Compression artifact reduction · Deep learning

1 Introduction

Video is widely employed in different fields, ranging from social media to self-driving cars. Although modern cameras can capture high-quality videos in many situations, there are some cases in which their quality is significantly reduced. For example, when videos are captured in poor light conditions or compressed to limit memory occupation, visible artifacts are introduced, causing problems to both user experience and many computer vision tasks.

✉ Claudio Rota
c.rota30@campus.unimib.it

Marco Buzzelli
marco.buzzelli@unimib.it

Simone Bianco
simone.bianco@unimib.it

Raimondo Schettini
raimondo.schettini@unimib.it

[1] Department of Informatics Systems and Communication, University of Milano - Bicocca, Viale Sarca 336, 20126 Milan, Italy

Video restoration aims to recover the clean video sequence from its degraded version. Different video restoration tasks can be defined: video denoising aims to remove noise, whose level can be high when videos are captured in particular imaging conditions; video deblurring aims to remove blur from videos that can be caused by out-of-focus subjects, moving objects, or camera shaking; video super-resolution aims to increase the spatial resolution of a given video to produce a high-resolution version of it from a low-resolution one; video compression artifact reduction aims to reduce artifacts introduced by compression algorithms that are applied to limit video memory occupation.

Many video restoration methods have been proposed in these years. They can be mainly divided into two categories: traditional methods and deep learning-based methods. In this paper, we focus our attention on deep learning methods because they represent an emerging category among the scientific community. We consider all the aforementioned restoration tasks to provide a global picture of the advances in video restoration because, although some methods are proposed to address a specific task, their building blocks and main features are not task-specific. In fact, some architectures were shown to be effective in different restoration tasks.

In this paper, we provide a comprehensive review of recent advances in video restoration using deep learning, analyzing the main features of some representative methods in an organized and structured manner, and highlighting their strengths and weaknesses. To the best of our knowledge, this is the first review of video restoration methods considering baseline schemes, architectural design strategies, convolution types, alignment techniques, and loss functions. Many surveys related to single-image restoration methods exist (Wang et al. 2020b; Tian et al. 2020a; Koh et al. 2021; Liu et al. 2020). Here we consider the video domain, which has been less investigated and presents several and different challenges. Recently, Liu et al. (2022) conducted a study on video super-resolution based on deep learning, focusing on alignment strategies. In this work, we extend the analysis to other video restoration tasks and to other aspects of video restoration methods.

Our main contributions can be summarized as follows:

- We provide a comprehensive review of existing video restoration methods based on deep learning, analyzing in a hierarchical manner their main features related to architectural choices, motion handling and loss functions, and discussing their advantages and limitations.
- We describe the characteristics of standard benchmark datasets, including their size in terms of number of sequences and frames, the resolution and the format of the contained video sequences, and classify them according to whether they contain synthetic or real distortions.
- We summarize the performance of the reviewed methods on the considered benchmark datasets, both in terms of effectiveness, reporting the corresponding information in terms of the standard metrics Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM), and efficiency, reporting the declared computational complexity and/or run time speed on given input resolutions and hardware configurations.
- We discuss the main challenges and future research directions in video restoration using deep learning, such as the need for real-time processing, improved strategies for frame alignment, multi-distortion restoration methods, better metrics and datasets, as well as the combination with traditional methods.

(a) Target frame. The shop sign is sharp while the person is blurry.

(b) Next frame. The shop sign is blurry while the person is sharp.

Fig. 1 Example of two consecutive frames containing different levels of distortions. Since in the target frame the person is blurry (red rectangle), some information in the next frame (green rectangle) can be used to restore the target frame, improving the final outcome. Images reprocessed from the REDS dataset (Nah et al. 2019a)

2 Background

Video restoration is the task that aims to remove artifacts introduced in videos by internal factors (e.g., noise) or external factors (e.g., camera shaking), producing a video of better quality. There is a huge variety of methods addressing the problem of video restoration. In recent years, research has been focused on the use of deep learning techniques. Therefore, this article reviews only methods in this category.

It is possible to see video restoration as a multi-image restoration task, where each video frame is restored using an image restoration method. However, this solution does not allow to exploit the temporal correlation among frames and may obtain suboptimal performance when the artifacts are strong, producing temporally inconsistent results because of the introduction of new temporal artifacts, such as flickering.

The main difference between image and video restoration methods is that the latter have the capability of using the temporal redundancy present in videos. Temporal redundancy means that the same information is contained within multiple frames, and video restoration methods can take advantage of this redundant information to recover details that may be missing in one frame. Indeed, neighboring frames typically contain the same objects, and such objects may appear with different levels of detail because of artifacts altering their aspect.

For instance, Fig. 1 shows two consecutive frames representing the same scene, but some contents in Fig. 1b appear sharper than in Fig. 1a. In such a case, temporal redundancy can be used to improve the quality of the results by aggregating sharper information from other frames. Even if neighboring frames contain the same objects, they may be located in different positions due to motion. Hence, an appropriate mechanism able to align frames is usually implemented.

The general framework of video restoration methods is reported in Fig. 2.

Given the target frame, video restoration methods take advantage of adjacent frames to obtain additional information useful to restore it. Typically, N previous and N subsequent frames are used to gather information both from the past and the future. Three modules with different purposes can be identified: (i) the alignment module is used to align input

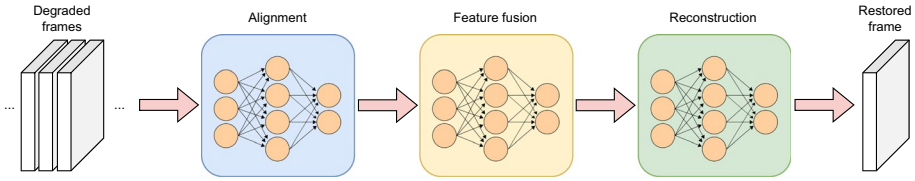


Fig. 2 General framework for video restoration methods. A sequence of adjacent frames is used as input. The alignment module aligns adjacent frames with the target one, the feature fusion module fuses the information contained in the aligned features, and the reconstruction module reduces the artifacts to produce the restored frame

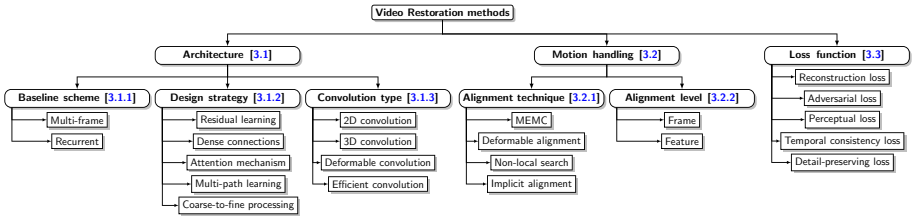


Fig. 3 Hierarchical organization of the features of video restoration methods reviewed in this paper. *MEMC* refers to motion estimation motion compensation

frames with the target one so that the obtained representations are spatially aligned; (ii) the fusion module aggregates the aligned representations and further refines them; (iii) the reconstruction module uses the fused representations to reduce artifacts and produce the restored frame. These modules can be implemented in different ways by different restoration methods, as discussed in the following sections.

3 Video restoration methods

Video restoration using deep learning is an active research field, and many methods have been proposed during these years. Although the differences among these methods may be quite large, they share some characteristics related to architectural choices, motion handling approaches and learning strategies. Therefore, instead of analyzing each method in isolation, we identify and review the main characteristics of video restoration methods and discuss their advantages and possible limitations. A graphical organization of the features analyzed in this paper is reported in Fig. 3. Table 1 provides a brief description of each feature analyzed, and summarizes its advantages and limitations, which are better clarified and motivated in the following.

3.1 Architectures

Defining the right neural architecture is one of the most critical problems in the field of video restoration, as it impacts the final performance both in terms of effectiveness and efficiency. In this section, we describe the two possible baseline schemes that can be used

Table 1 Brief description and summary of the main advantages and limitations of the video restoration features analyzed according to the hierarchical organization in Fig. 3

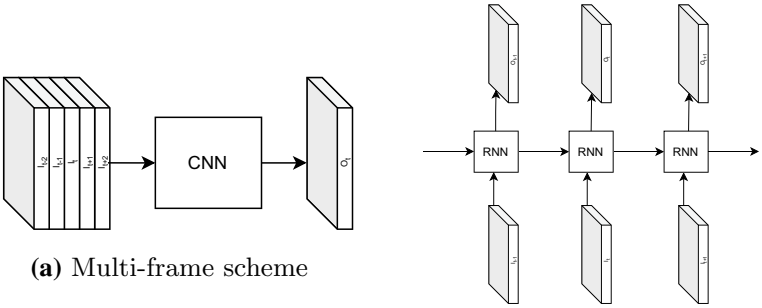
			Short description	Advantages	Limitations
Architecture	Baseline scheme	Multi-frame	Multiple frames are stacked and used as input	Easy to implement	Limited temporal context The same frame is processed multiple times The number of input frames is a hyper-parameter to tune
		Recurrent	Each frame is processed sequentially and temporal information is aggregated in hidden states, as in RNNs	Efficiency Long temporal context	Suitable mechanisms to aggregate information required
	Design strategy	Residual learning	Skip connections propagate the input directly to the output and the network learns the residual	Learning the direct transformation of the input is easier (G) Vanishing gradient mitigated (L)	–
		Dense connections	Dense blocks with skip connections that forward the output of each layer to the input of subsequent layers are introduced	Richer patterns can be learned Receptive field increased Vanishing gradient mitigated	–
		Attention mechanism	Specialized modules are designed to weight features according to their importance	Networks can distinguish between relevant and irrelevant features	Computational complexity
		Multi-path learning	Multiple and separated paths are used to process different aspects of the input	Modeling capabilities potentially increased (G) Input is analyzed using multiple receptive fields (L)	–
		Coarse-to-fine processing	Input is analyzed in multiple resolutions to focus on different levels of detail	Feature reuse avoids repeated computations Suitable to handle large motion	Small fast moving objects in coarser levels may not be detected
	Convolution type	2D	Filters can move along two dimensions, i.e., height and depth	–	Limited capability in modeling temporal dependencies
		3D	Filters can move along three dimensions, i.e.,height, width and depth	Both spatial and temporal dependencies can be modeled	Computational complexity
		Deformable	2D offsets are added to deform the rigid sampling grid of standard convolutions	Receptive field adapts to the input Only relevant input information is considered	Computational complexity Training instability due to offset overflow

Table 1 (continued)

			Short description	Advantages	Limitations
		Efficient	Efficient version of standard convolutions that requires less parameters and operations	Efficiency	Limited accuracy
Motion handling	Alignment technique	MEMC	Motion vectors between target and adjacent frames are estimated and later used to align them	Accurate alignment when motion estimation is correct Self-supervised training can be used	Errors in motion estimation lead to alignment artifacts Sensitive to luminance changes, fast motions and occluded objects Computational complexity
		Deformable alignment	Deformable convolutions are used to capture motion cues and produce aligned features	Receptive field is dynamically adjusted End-to-end training with the rest of the framework Multiple spatial offsets for each pixel increase robustness to luminance changes and occlusions	Training instability due to offset overflow Computational complexity
		Non-local search	Pixel/patch similarity is computed between target and adjacent frames to detect regions referring to the same objects	Global receptive field Distant pixels can contribute to the alignment process Robust to large motion magnitude	Efficiency The dimension of search area is a hyperparameter
		Implicit alignment	No specific modules are used for alignment and the network learns to find the information it needs by itself	No need of ad-hoc modules for alignment Artifacts related to wrong flow estimation are prevented	Fixed receptive field Ineffective alignment when motion is too large
	Alignment level	Frame	Alignment is performed on input frames	Self-supervised training can be used Results areinterpretable	Loss of frame details
		Feature	Alignment is performed on features extracted from input frames	More accurate alignment	Results are not interpretable
Loss function		Reconstruction loss	Pixel-wise error between output and ground truth	Main guide for the learning process	Perceptually unsatisfying results may be produced
		Adversarial loss	A discriminator network is added to judge how much realistic the results of the restoration network are	More realistic results Useful to suppress new artifacts introduced by the network	Possible training instability The produced results may be different from the expected ones Regularization term required

Table 1 (continued)

		Short description	Advantages	Limitations
	Perceptual loss	Loss focusing on improving perceptual similarity	Perceptual similarity improved	Possible training instability Training time increased Regularization term required
	Temporal consistency loss	Loss focusing on improving temporal coherence of the results	Temporal consistency imposed via loss function and learned during training	Regularization term required Training time may be increased Redundant computations or modifications of the network output required for multi-frame methods
	Detail-preserving loss	Loss focusing on recovering details and textured regions	Details and textures are improved Oversmoothing reduced	Regularization term required



(a) Multi-frame scheme

(b) Recurrent scheme

Fig. 4 Schematic representation of the main baseline schemes

by video restoration methods. Then we review the main design strategies and the convolution types used to model the spatial and temporal relationships among frames.

3.1.1 Baseline schemes

Video restoration methods take advantage of temporal redundancy to access the information contained in the temporal neighborhood. To this end, two baseline schemes can be used, i.e., the multi-frame and the recurrent approaches, that are schematically represented in Fig. 4.

3.1.1.1 Multi-frame The simplest strategy to give the network access to temporal information is the multi-frame approach. It consists in using a temporal sliding-window of a fixed size centered on the target frame. The target frame and its neighboring frames are stacked and this represents the input to the restoration methods, as shown in Fig. 4a. The dimension of the temporal window is a hyperparameter to tune and is usually set between three (Caballero et al. 2017; Guan et al. 2019; Claus and Gemert 2019) and seven (Jo et al. 2018; Xue et al. 2019; Deng et al. 2020). A too small window may prevent the network from fully exploiting the potential information in the temporal neighborhood (Zhang et al. 2018a; Haris et al. 2019), whereas a too large window increases the computational complexity (Claus and Gemert 2019) and may include frames containing irrelevant information due to large object motion (Zhang et al. 2018a). Methods based on the multi-frame scheme usually process a frame multiple times, depending on the window size, and this might result in a waste of computational resources. Although these limitations can be addressed by using different strategies, the multi-frame scheme is widely employed also by recent methods (Paliwal et al. 2021; Chen et al. 2021; Vaksman et al. 2021), as it is a simple yet effective solution to exploit temporal context.

3.1.1.2 Recurrent An alternative solution to capture information from the temporal context is the use of Recurrent Neural Networks (RNNs). Following this approach, illustrated in Fig. 4b, each frame is progressively passed through the network that extracts its features, aggregates them into a hidden state to be used for future frames, and uses relevant information from the previously processed frames to restore it. Methods using the recurrent scheme are usually faster than the ones based on multi-frame because each frame is only

processed once, and can potentially achieve better performance because they can exploit a larger temporal window. However, they require suitable mechanisms to aggregate the features extracted from multiple frames. To this end, different strategies have been proposed (Hyun Kim et al. 2017; Nah et al. 2019b; Zhong et al. 2020; Zhou et al. 2019; Isobe et al. 2020). Hyun Kim et al. (2017) developed a strategy to blend feature maps of previous frames and the ones of the current frame by using a convolutional layer. Nah et al. (2019b) realized an iterative procedure using the outputs of RNN cells as inputs to the same cell multiple times. Isobe et al. (2020), inspired by Dynamic Filter Networks (Jia et al. 2016), implemented a module to adapt the hidden state to the appearance of the current frame by using correlation to highlighting only the most similar features.

An important aspect of recurrent methods is how the information is propagated through the framework. Usually, it is propagated from the initial frame to the last one (Nah et al. 2019b; Zhong et al. 2020; Zhou et al. 2019; Hyun Kim et al. 2017; Zhao et al. 2021). Such unidirectional propagation may result to be suboptimal because the amount of information received when processing different frames is different, as the first frames have access to less information than the last ones. Some methods (Huang et al. 2017b; Chan et al. 2021a, 2022; Zhu et al. 2022) use bidirectional information propagation, where information is propagated both forward and backward so that each frame can also benefit from the information coming from subsequent frames. Chan et al. (2021a) conducted a study demonstrating that bidirectional propagation improves the restoration performance.

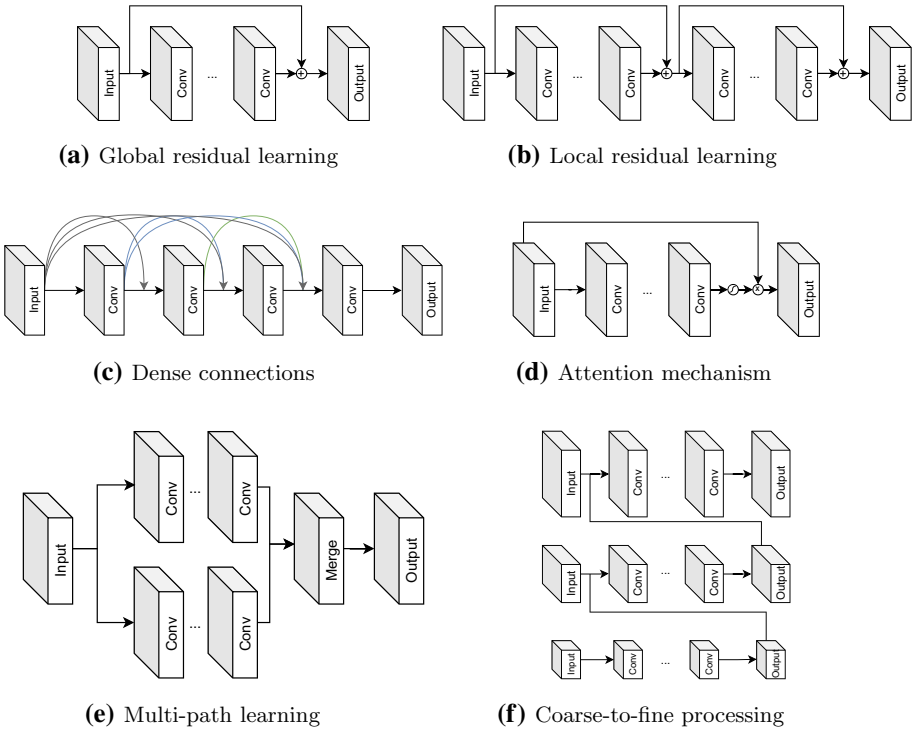
3.1.2 Design strategies

When designing a deep neural network, there are several issues that one has to deal with. For instance, deep architectures suffer from the vanishing gradient problem, which can degrade the performance by preventing layers close to the input to be properly optimized. Another issue is feature modulation, since not all the features extracted by neural networks carry information that are actually useful for the considered task. To tackle these problems, several strategies are proposed and used by researchers to build their networks by combining them in different ways. Figure 5 reports the most common architectural design strategies, which are analyzed in detail in the following.

3.1.2.1 Residual learning He et al. (2016) proposed ResNet and demonstrated that residual learning can facilitate the training process and improve accuracy for image classification. Then, it has been widely adopted for other computer vision tasks, including video restoration. There are two possible implementations of residual learning to design a CNN: global and local residual learning.

Global residual learning is used to model situations where the output is highly correlated with the input, such that it is easier to learn a direct transformation of the input rather than a deep one. It is usually realized by adding a skip connection from the input to the output, so that the network only needs to learn, for example, the difference between input and output (Su et al. 2017; Guan et al. 2019; Zhou et al. 2019; Wang et al. 2019; Deng et al. 2020), as shown in Fig. 5a.

Local residual learning is primarily used to mitigate the vanishing gradient problem, and it consists in using blocks composed of groups of convolutions with skip connections, as in ResNet or variants of it (Zhang et al. 2018a; Nah et al. 2019b). Some works (Nah et al. 2017; Lim et al. 2017) empirically experimented that slight modifications of the



(a) Global residual learning **(b)** Local residual learning

(c) Dense connections **(d)** Attention mechanism

(e) Multi-path learning **(f)** Coarse-to-fine processing

Fig. 5 Schematic representation of the common architecture design strategies

original ResNet block were beneficial for the restoration performance. Figure 5b illustrates the design of a network adopting local residual blocks.

3.1.2.2 Dense connections Dense connections have quickly spread since the introduction of DenseNet (Huang et al. 2017a). A dense block is characterized by several skip connections that forward the output of each layer directly to the input of the subsequent layers, so that each layer receives collective knowledge from all the previous layers. Figure 5c shows the architecture design of a network with dense connections.

Similar to residual learning, the use of dense connections is beneficial for the vanishing gradient problem. In addition, it allows feature reuse making it possible to learn richer patterns, it allows to increase the network receptive field, and it allows to use networks with fewer parameters because dense blocks have a relatively small growth rate, i.e., the additional number of channels for each layer. Some methods (Guan et al. 2019; Jo et al. 2018) adopted dense connections observing an improvement in the overall restoration results. Zhong et al. (2020) integrated dense blocks within RNN cells mainly to reduce the computational complexity of their model.

3.1.2.3 Attention mechanism Attention mimics cognitive attention, defined as the ability to choose and concentrate mainly on relevant stimuli. In computer vision, the attention mechanism can be considered as a dynamic selection process that is realized by weighting features according to their importance in producing the output. Figure 5d shows a general

implementation of the attention mechanism, where a sigmoid activation is used to produce weights between 0 and 1 and element-wise multiplication is used to modulate the input by suppressing irrelevant features. Typical attention types are: (i) channel attention, used to select the most important channels; (ii) spatial attention, used to select the most important regions; (iii) temporal attention, used to select the most important frames (Guo et al. 2022).

Wang et al. (2019) used temporal attention to identify the frames most similar to the target one, and spatial attention to mitigate errors arising from wrong frame alignment. Mehta et al. (2021) inserted the channel attention module proposed in SqueezeNet (Hu et al. 2018) into their network to better model dependencies across channels. Similarly, Zhong et al. (2020) adopted the same module but with slight modifications to improve the fusion of features from past and future frames. Zhao et al. (2021) designed a spatial attention module using deformable convolutions (Zhu et al. 2019) to highlight artifact-rich areas in each frame, such as boundary areas of moving objects, so that their model can focus more on removing artifacts in such areas. Paliwal et al. (2021) combined channel and spatial attention to identify errors related to optical flow computation, such as occlusions, by using SqueezeNet blocks and Convolution Block Attention Modules (CBAM) (Woo et al. 2018).

Designing architectures with attention modules can increase the overall effectiveness because they allow to distinguish relevant features from irrelevant ones and to weight them accordingly. The main disadvantage is related to the efficiency, since including attention leads to an increase in the number of parameters and operations.

3.1.2.4 Multi-path learning Multi-path learning refers to processing features using multiple and separate paths that finally merge the complementary information. Multi-path learning can be either global or local.

In the global version, multiple parallel paths focus on different aspects of the input, as shown in Fig. 5e. Usually, two separate paths are used by video restoration methods (Jo et al. 2018; Chen et al. 2021; Isobe et al. 2020; Zhou et al. 2019). Jo et al. (2018) used one path to learn upscaling filters (Jia et al. 2016) and the other to learn high-frequency components, with the two paths sharing most of the weights. Isobe et al. (2020) separated low-frequency and high-frequency components, i.e., structures and details in the spatial domain, and processed them using separate branches. Chen et al. (2021) proposed a two-branch network with independent weights, where one branch is used to extract spatial features from individual frames and the other one to extract temporal features from multiple frames. These features are finally merged using a stack of convolutions.

Local multi-path learning is inspired by Inception modules (Szegedy et al. 2015), which are composed of multiple paths containing convolutions with different kernel sizes to analyze the input using multiple receptive fields. Mehta et al. (2021) included local multi-path learning in their network using layers composed of three convolutions with filters of size 3×3, 5×5 and 7×7, whose results are finally summed up. Zhao et al. (2021) employed two local branches with different receptive fields to increase the accuracy of offset prediction for deformable convolutions (Zhu et al. 2019).

While global multi-path learning can provide better modeling capabilities, as there are multiple paths focusing on improving different aspects of the input, local multi-path learning allows to extract multi-scale features by looking at the input with multiple receptive fields.

3.1.2.5 Coarse-to-fine processing In visual recognition, coarse-to-fine processing refers to applying a method to a downscaled version of the image, i.e., coarse, and then gradually

increasing its resolution and propagating the results to the fine version. In a coarse-to-fine architecture, as illustrated in Fig. 5f, the input is downscaled multiple times and processed by the network starting from the coarsest level (i.e., the lowest resolution), and the output is first upscaled and then propagated to the upper level until the finest level (i.e., original resolution) is reached. The main idea behind this approach is that the network can process the main structures at the coarsest level, while focusing on the details at the finest level. Propagating the outputs to upper levels allows the network to reuse features from lower levels, avoiding repeated computations and focusing on higher-level abstractions. All the levels usually share the same structure.

The coarse-to-fine design is typically adopted in the context of optical flow estimation, where it is known to be an effective solution for modeling large motion between objects and improving the estimation accuracy (Amiaz et al. 2007). Some methods for video restoration (Caballero et al. 2017; Guan et al. 2019; Yang et al. 2018) developed a coarse-to-fine module for motion estimation and compensation, which starts by computing optical flow at the lowest resolution and then propagates the estimated flow to upper levels for refinement, whereas others (Xue et al. 2019; Chan et al. 2021a, 2022) integrated an existing coarse-to-fine network (Ranjan and Black 2017) inside their framework to increase flow estimation accuracy.

A limitation of this approach is that coarse-to-fine networks may struggle in detecting small fast moving objects in coarse levels because they are removed by downscaling operations, and thus they are not suitable to handle large motion in this case (Savian et al. 2020).

3.1.3 Convolution types

In video restoration, both spatial and temporal correlations among neighboring frames require to be properly modeled to produce detail-rich and temporally-coherent results. To this end, different convolution types can be used.

3.1.3.1 2D convolutions 2D convolutions are the most commonly used type, which consists in centering a 2D filter on each spatial element of the features and then summing up the element-wise product between element neighbors and filter weights. The 2D convolution transforms a 2D matrix of features into a different 2D matrix of features that is passed as input to the next layer. Video restoration methods use 2D convolutions to process and fuse features coming from multiple input frames. The first convolutional layer typically fuses all the frames, and the next layers have only a limited effect in modeling additional temporal information because, after the application of the first layer, the temporal dimension is squeezed and later convolutions only operate on the spatial dimension (Fan et al. 2019). Therefore, 2D convolutions are effective in abstracting spatial dependencies, but they are not fully adequate in capturing temporal ones.

3.1.3.2 3D convolutions A solution to take into account the temporal correlation among frames is the use of 3D convolutions. The main difference is that filter depth and input depth in 3D convolutions are not constrained to be equal as in 2D convolutions. Thus, a 3D filter can move in all the three dimensions, i.e., height, width, and depth. At each position, the element-wise product and addition produce one number, hence the output is a 3D data structure. 3D convolutions can capture spatial relationships in the input data, as 2D convolutions do, but they can model temporal relationships as well (Tran et al. 2015). While Zhang et al.

(2018a) employed only 3D convolutions, other methods (Jo et al. 2018; Chen et al. 2021; Vaksman et al. 2021) used 3D convolutions together with 2D ones to better handle spatial and temporal information. The main limitation of 3D convolutions is related to efficiency, since applying them increases the number of operations to perform.

3.1.3.3 Deformable convolutions Deformable convolutions were introduced by Dai et al. (2017) to address the limited capability of CNNs in modeling large and unknown transformations, originated by the rigid sampling grid of standard convolutions. In deformable convolutions, 2D offsets are added to the regular grid sampling locations, deforming the constant receptive field of the standard convolution operation. For each location, the applied deformation depends on the input features: the offsets are computed from the input feature map using additional convolutional layers, whose weights are learned during training. Zhu et al. (2019) proposed an enhanced version of deformable convolutions, where modulation scalars, i.e., position-specific weights used to modulate the weights of each convolution operation, are learned along with 2D offsets. In video restoration, deformable convolutions are typically used for frame alignment (Wang et al. 2019; Tian et al. 2020b; Deng et al. 2020; Yue et al. 2020; Chan et al. 2022; Zhao et al. 2021). Using deformable convolutions, a network can adapt its receptive field according to object scales, being so able to handle the large pixel displacement caused by motion. However, additional parameters representing the 2D offsets and modulation scalars must be learned during training.

3.1.3.4 Efficient convolutions Model efficiency is crucial in real-time applications. A possible solution to reduce model complexity is to replace standard convolutions with more efficient learnable layers, such as separable and depth-wise convolutions (Chollet 2017; Howard et al. 2017; Mehta et al. 2021; Xiao et al. 2021; Vaksman et al. 2021).

Separable convolutions exploit the separability of the standard convolution operation along the spatial dimensions, so that a two-dimensional kernel can be separated into two one-dimensional kernels, reducing the number of parameters. However, since not all kernels can be separated, the use of separable convolutions may degrade the performance.

In depth-wise convolutions, each input channel is convolved with each kernel channel, but instead of summing them up as in standard convolutions, the output channels are simply stacked together. These kinds of convolutions were introduced to increase efficiency because the total number of operations to perform is lower than the one in regular convolutions, but they also may lead to a decrease in performance (Bao et al. 2020).

3.2 Motion handling

Motion is an intrinsic characteristic of video data. Video restoration methods must deal with it if they want to be able to exploit spatial information of adjacent frames. In this section, we first analyze the main alignment techniques used by video restoration methods to align neighboring frames with the target one, then we discuss why some methods perform alignment at feature level instead of at frame level.

3.2.1 Alignment techniques

Alignment techniques are used by video restoration methods to spatially align adjacent frames with the target one, so that information referring to the same objects in multiple

frames will be located at the same spatial positions, and it will be aggregated and accessed more easily. Many solutions to align frames were proposed and can be grouped in a few categories, as reported in the following. The decision of using an alignment technique instead of another one is important for a video restoration method since it can have a measurable impact on the final performance, as some studies demonstrated (Chan et al. 2021a, b; Zhou et al. 2022).

3.2.1.1 Motion Estimation Motion Compensation (MEMC) The most common technique for handling motion in video restoration is the Motion Estimation Motion Compensation (MEMC) approach (Xue et al. 2019). This solution aligns frames in two steps: first, it performs motion estimation, which aims to estimate per-pixel motion between a source and a target frame, and then applies motion compensation, which aims to warp the source frame to the target one according to the estimated motion. Motion estimation is typically done by optical flow computation (Beauchemin and Barron 1995), which is the task that computes per-pixel motion vectors between two frames. Given the source frame I_s and the target frame I_t, the flow map $F_{s\to t}$ describing how pixels moved can be defined as:

$$F_{s\to t} = ME(I_s, I_t) \tag{1}$$

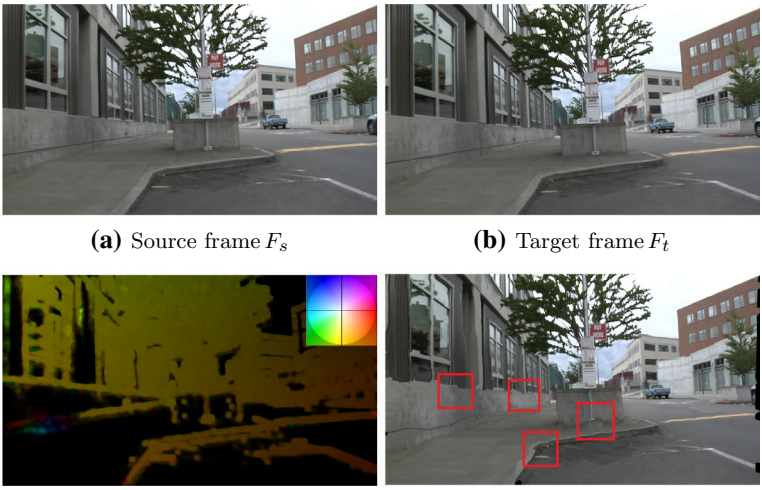
where ME is the motion estimation operation. Motion compensation shifts the pixel positions in the source frame I_s according to the per-pixel vectors contained in the flow map $F_{s\to t}$. The warped frame $\hat{I}_t$ is obtained as:

$$\hat{I}_t = MC(I_s, F_{s\to t}) \tag{2}$$

where MC is the warping operation that can be implemented by using bilinear interpolation or the sampling layer of a Spatial Transformer Network (STN) (Jaderberg et al. 2015).

Optical flow computation was originally defined as a handcrafted optimization problem (Weinzaepfel et al. 2013; Revaud et al. 2015; Hu et al. 2017), but the growing spread of deep learning has led to the development of CNN-based models that can produce more accurate results than traditional methods (Dosovitskiy et al. 2015; Ranjan and Black 2017; Sun et al. 2018; Teed and Deng 2020). Some video restoration methods (Xue et al. 2019; Pan et al. 2020; Chan et al. 2021a; Son et al. 2021; Paliwal et al. 2021) directly integrated an existing CNN-based method for optical flow estimation within their architectures. Xue et al. (2019) adopted SpyNet (Ranjan and Black 2017) as flow estimation network and STN (Jaderberg et al. 2015) to perform frame warping, while Chan et al. (2021a) used the same model but opted for plain bilinear interpolation. In contrast, Pan et al. (2020) employed PWC-Net (Sun et al. 2018), Paliwal et al. (2021) used RAFT (Teed and Deng 2020), whereas Son et al. (2021) adopted LiteFlowNet (Hui et al. 2018) due to its efficiency. Other methods (Caballero et al. 2017; Yang et al. 2018; Guan et al. 2019) developed their own modules to perform frame alignment using MEMC. Caballero et al. (2017) built a Spatio-Temporal Motion Compensation (STMC) module, adopting a coarse-to-fine processing approach that propagates coarser flows to upper levels for progressive refinements. Due to excessive downscaling, the accuracy of the estimated motion vectors was reduced. Therefore, Yang et al. (2018) and Guan et al. (2019) later improved upon STMC by introducing an additional flow estimation layer without any downscaling operation.

Existing optical flow estimation methods do not expect to receive degraded frames as input, hence a retraining procedure is necessary for the adaptation to the considered task, typically using pretrained models as starting point. Accurate ground truth for optical flow estimation cannot be obtained, unless a dataset is synthetically generated. A possibility is



(a) Source frame F_s **(b)** Target frame F_t

(c) Optical flow $F_{s\rightarrow t}$ **(d)** Warped frame $\hat{F}_t$

Fig. 6 Example of motion estimation and compensation between two frames. In the warped frame $\hat{F}_t$, motion estimation and compensation artifacts are visible (red squares). Black pixels on the right-hand side of $\hat{F}_t$ are due to occlusions. Images reprocessed from the DVD dataset (Su et al. 2017)

to estimate flow maps using pretrained models on ground truth frames and use the obtained maps as ground truth to adapt flow estimation methods to degraded frames. However, the domain gap between datasets for optical flow methods and video restoration methods may lead to inaccurate flow estimations (Son et al. 2021). Therefore, a common solution is represented by self-supervised training, where the model is used to compute optical flow between two frames, warping operation is performed to align them according to the estimated flow, and a warping loss is employed to guide the learning procedure (Caballero et al. 2017; Xue et al. 2019; Pan et al. 2020; Son et al. 2021; Paliwal et al. 2021).

The MEMC strategy for motion handling is widely used by video restoration methods and has multiple advantages and disadvantages. Accurate flow map prediction enables accurate alignment, making the process of information extraction and fusion easier because information referring to the same objects in multiple frames are located in the same spatial locations. In addition, self-supervised learning represents an effective training strategy to adapt models to compute optical flow even on frames affected by artifacts when ground truth flow maps are not available. However, when videos contain luminance changes, fast motion, or occluded objects, the performance of methods based on MEMC alignment may considerably degrade (Savian et al. 2020). Errors in flow map prediction imply errors in frame alignment, introducing new artifacts that damage the entire restoration process (Tassano et al. 2020). Figure 6 shows an example of artifacts introduced by wrong motion estimation. To address this problem, different solutions were proposed (Tassano et al. 2019; Paliwal et al. 2021; Son et al. 2021). Tassano et al. (2019) suggested to preprocess input frames individually using a CNN with the aim of removing part of the artifacts before flow estimation, because optical flow is highly sensitive to noise. In contrast, Paliwal et al. (2021) postprocessed warped frames using residual modules (Zamir et al. 2020) with attention mechanisms (Hu et al. 2018; Woo et al. 2018) to discard artifacts introduced by MEMC errors. Son et al. (2021) provided multiple alignment candidates so

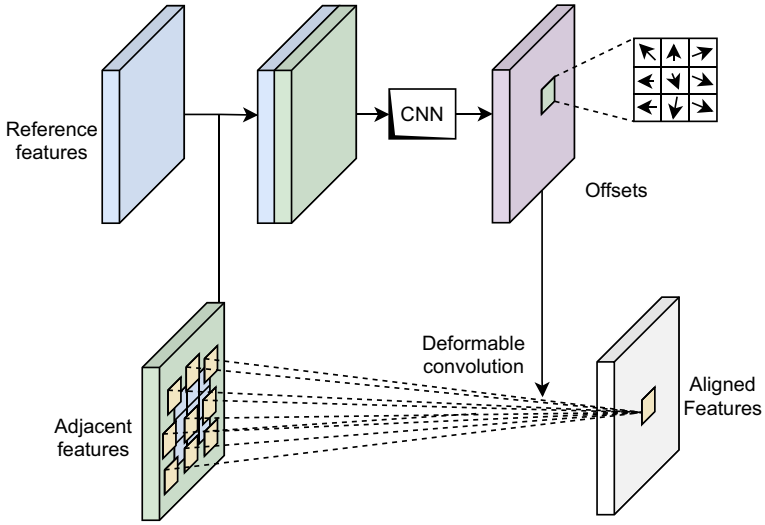


Fig. 7 Deformable alignment. Reference features and features from adjacent frames are processed together to estimate position-specific spatial offsets to deform the rigid sampling grid of standard convolutions

that the network can leverage on multiple alignment solutions and find the most appropriate one. Another drawback in using MEMC for frame alignment is related to the computational complexity, since the estimation of per-pixel flow maps and the warping operation on high-resolution frames considerably impact the overall complexity (Bovik 2009).

3.2.1.2 Deformable alignment Deformable convolutions in video restoration were introduced as an alternative strategy to align frames without the need to explicitly compute the optical flow between them (Tian et al. 2020b). Depending on the input, the network can decide the best transformations to apply to obtain aligned features, from which it will extract the information needed to restore the target frame.

Different implementations of deformable alignment exist (Tian et al. 2020b; Wang et al. 2019; Chan et al. 2022; Deng et al. 2020). The general framework is illustrated in Fig. 7. Given the features extracted from the target frame and the ones extracted from an adjacent frame, they are initially fused (e.g., by concatenation) and then processed by a CNN to estimate the deformable offsets that will be used to deform the sampling grid of the standard convolution used to process, and consequenty align, the features of the adjacent frame. As a result, since deformable convolutions can capture motion cues, the produced features will be spatially aligned with the reference ones.

Tian et al. (2020b) were the first to propose deformable alignment in video restoration, adopting an alternating sequence of regular convolutions for deformable offset estimation and deformable convolutions to perform alignment. Inspired by this work, Wang et al. (2019) developed a deformable alignment module implementing a coarse-to-fine processing approach that propagates the learned offsets from lower levels to upper ones, progressively increasing offset accuracy. A similar solution was later used by Yue et al. (2020). These solutions perform deformable alignment in a pairwise manner, i.e., computing the deformable offsets by taking into account the target frame and only one of its adjacent frames at a time, thus failing to fully exploit temporal correlations among multiple frames. To address this limitation, some methods (Deng et al. 2020; Zhao et al.

Fig. 8 Example of similar patches in consecutive frames. Using non-local search, video restoration methods can localize matching patches in multiple frames and use them to produce aligned features. Images reprocessed from the BSD dataset (Zhong et al. 2020)

2021; Xu et al. 2021) adopted an encoder-decoder architecture to predict deformable offsets by jointly processing the entire stack of frames, better exploiting temporal correlations among frames and also increasing offset prediction accuracy due to the large receptive field of encoder-decoder architectures.

Using deformable alignment instead of MEMC for handling motion brings multiple advantages. While in optical flow only one spatial offset for each spatial location is estimated, deformable convolutions learn multiple and complementary offsets (e.g., nine in the case of a 3×3 kernel) that can mitigate the problem of occlusions and reduce errors caused by large motion (Chan et al. 2021b). Deformable alignment is also less sensitive to varying illumination and motion conditions than the MEMC approach. Moreover, the module for deformable alignment can be trained together with the restoration framework in an end-to-end manner, without requiring any adaptation as in MEMC. The main issue in using deformable alignment is related to the training process, which may suffer from instability due to offset overflow, degrading the overall performance of the models (Chan et al. 2021b). Chan et al. (2022) tackled this issue by designing a flow-guided deformable alignment scheme, where optical flow is used to guide the deformable alignment. More precisely, they employed optical flow to warp features from the previous frame to the target ones and used them to predict offsets for deformable convolutions.

3.2.1.3 Non-local search In video restoration, non-local search represents an alignment strategy mainly introduced to obtain a global receptive field, thus overcoming the limitation of convolution operations that perform computations in local areas. The main idea behind this approach is to allow even distant pixels to contribute to the alignment process regardless of motion magnitude. The goal of non-local search is to find pixels within the adjacent frame that are most similar to the ones in the target frame, and use them to perform alignment. Computing pixel similarity between two frames allows to detect region patches belonging to the same objects, whose similarity is expected to be high. Figure 8 shows an example of similar patches in three adjacent frames. Using non-local search, video restoration methods can compute pixel similarity to find matching region patches and combine them to perform frame alignment.

Several methods using non-local search to handle object motion have been proposed (Yi et al. 2019; Xu et al. 2019; Li et al. 2020a; Davy et al. 2019; Vaksman et al. 2021). While some methods (Yi et al. 2019; Xu et al. 2019; Li et al. 2020a) integrate non-local search within their network as a learnable component, others (Davy et al. 2019; Vaksman et al. 2021) employ it to generate aligned frames to use as inputs to their CNNs by adopting a handcrafted procedure. Inspired by non-local networks (Wang et al. 2018),

Yi et al. (2019) computed pixel correlation between each pixel of the target frame and all the pixels of adjacent frames, then they generated output pixels by performing a weighted sum of pixels of adjacent frames using correlations as weights. Xu et al. (2019) included non-local search within ConvLSTM modules (Xingjian et al. 2015). They computed the similarity between the pixels of the current frame and all the pixels of the previous frame to generate a similarity matrix, which is later used to update the ConvLSTM outputs. Instead of working at pixel level, Li et al. (2020a) locally selected the top-K patches in the adjacent frames that are most correlated with a given patch in the reference frame. They are sorted according to their similarity, fused using convolutional layers, and used to generate aligned feature maps. Davy et al. (2019) proposed a non-local search to produce aligned feature maps to use as input to their CNN. For each pixel in the target frame, they centered a patch on it and searched for similar patches in the temporal neighborhood. Then, they sorted these patches and created a vector containing the central pixels of each patch. Since the use of only central pixels does not allow to properly consider the spatial dependencies among pixels, thereby limiting the alignment effectiveness, Vaksman et al. (2021) crafted different versions of the target frame by directly aggregating patches from adjacent frames. After finding all the possible overlapping patches of the target frame, they searched for the most similar patches in the adjacent frames for each of them. Then, they created different versions of the target frame by stitching non-overlapping patches together, starting from the most similar ones.

Methods adopting non-local search are less sensitive to motion magnitude, since arbitrarily distant pixels can be involved in the alignment process. Their main drawback is related to the increase in computational complexity caused by the computation of pixel similarity. Some methods (Davy et al. 2019; Vaksman et al. 2021; Li et al. 2020a) addressed this problem by limiting the search area, which becomes a hyperparameter to tune. Instead, Xu et al. (2019) proposed to reduce the frame spatial dimension using pooling operations before computing pixel similarity, at the cost of reduced accuracy.

3.2.1.4 Implicit alignment Methods adopting implicit alignment do not include any specific module for frame alignment, but they rely on the capability of the networks to learn proper transformations that allow them to make the most of the information shared across frames. The key element in implicit alignment is the network layer receptive field that has to be large enough to cover possible pixel displacement to accurately align frames. Convolutions have a receptive field restricted to the kernel size, which is typically constrained between 3×3 and 7×7. A common solution to enlarge the receptive field is to stack convolutions and to use pooling operations. Thus, video restoration methods usually implement encoder-decoder architectures, in which the encoder typically contains pooling operations (Su et al. 2017; Zhou et al. 2019; Wang et al. 2020a; Tassano et al. 2020; Chen et al. 2021), or adopt residual blocks, which contain stacked convolutions (Zhang et al. 2018a; Nah et al. 2019b; Isobe et al. 2020).

Su et al. (2017) demonstrated that state-of-the-art performance could be obtained by using implicit alignment, developing an encoder-decoder architecture to extract and fuse information from multiple frames. Later, Nah et al. (2019b) proposed to combine an encoder-decoder architecture with residual blocks at the bottleneck, and to insert it within RNN cells for both frame restoration and hidden state update. Zhong et al. (2020) used a similar approach, replacing residual blocks with dense blocks and adding attention modules (Hu et al. 2018) for feature reweighting. Tassano et al. (2020) proposed to cascade

two encoder-decoders, developing a two-stage architecture to avoid flow-related artifacts. A similar approach was later adopted by Wang et al. (2020a), who used the same two-stage architecture but preceded by an encoder-decoder to restore single frames before the aggregation. Some methods (Jo et al. 2018; Zhang et al. 2018a; Chen et al. 2021) also included 3D convolutions for better motion handling, since these are more suitable to model video data because they can also move along the temporal dimension. Jo et al. (2018) combined 2D and 3D convolutions within dense blocks, developing a dense residual network. Zhang et al. (2018a) adopted only 3D convolutions, integrating them in residual blocks and cascading multiple modules. In contrast, Chen et al. (2021) used an encoder-decoder architecture with 3D convolutions to generate aligned features, while using a parallel network with 2D convolutions to obtain only spatial information from single frames. Zhou et al. (2019) proposed to enrich an encoder-decoder with a Filter Adaptive Convolutional (FAC) module that assigns position-specific weights to regular convolutions, as objects in the scene do not have the same motion and should be treated accordingly.

Using implicit alignment allows to prevent artifacts related to wrong motion estimation, typically introduced by methods using the MEMC technique. In addition, it avoids the need of designing ad-hoc modules for frame alignment because the burden of finding suitable frame transformations is entirely left to the network. However, the lack of dedicated mechanisms for alignment might make it difficult to properly align features, especially in presence of large motion, because of the fixed and limited receptive field of convolutions, which could not have access to a context large enough to properly combine the information coming from the input frames (Chan et al. 2021a). Enlarging the receptive field by stacking convolutions quickly increases the computational complexity, while using pooling operations may remove important details.

3.2.2 Alignment levels

Different alignment techniques can be adopted to align adjacent frames with the target one. These strategies can be applied either directly to input frames or to features extracted from them.

3.2.2.1 Frame level Alignment at frame level is typically adopted by methods using the MEMC alignment strategy. Indeed, several methods perform alignment by computing optical flow and warping frames before the actual restoration process (Caballero et al. 2017; Xue et al. 2019; Yang et al. 2018; Guan et al. 2019; Pan et al. 2020; Paliwal et al. 2021). The warping operation is directly applied to adjacent frames to align them with the target one for later processing. However, spatial warping introduces information loss on frame details because of the interpolation operation required to handle fractional flow offsets (Chan et al. 2021b). Chan et al. (2021a) experimented that performing alignment at frame level using optical flow may also introduce blurriness and other types of artifacts. Some methods based on deformable alignment strategy (Deng et al. 2020; Zhao et al. 2021; Xu et al. 2021) apply deformable convolutions to input frames to produce aligned feature maps later used as inputs to their restoration networks. Similarly, some methods (Davy et al. 2019; Vaksman et al. 2021) applied non-local search to input frames to create multiple frame versions to be fed to their restoration networks. The main advantage of performing alignment at frame level is the possibility of using self-supervised training, where alignment can be directly guided via loss functions imposed between

aligned and reference frames. Moreover, with this approach the interpretability of the alignment phase is increased, allowing a straightforward inspection of the results.

3.2.2.2 Feature level Instead of directly trying to align frames, an alternative solution is to align features extracted from them. All the methods adopting an implicit alignment strategy perform alignment at feature level by progressively applying feature transformations. Chan et al. (2021a, 2021b) conducted a study on the impact of moving the alignment phase from frame to feature level, showing that the latter improved the performance. This outcome motivated the development of some video restoration methods (Chan et al. 2021a, 2022), which adopt a MEMC alignment strategy with optical flow estimated and applied to features rather than to frames. Similarly, some methods adopting deformable alignment (Tian et al. 2020b; Wang et al. 2019; Yue et al. 2020) apply deformable convolutions to features maps instead of frames. In this case, an encoder is used to extract features from frames before alignment, and deformable convolutions are applied to them. The key advantage of feature alignment is that it leverages the capability of neural networks to learn the most suitable internal representations of input frames to make the alignment process easier and more accurate. Besides, alignment at feature level makes models more robust to noise (Sun et al. 2018).

3.3 Loss functions

Loss functions are used in training to quantify the error made by the network in the forward pass. Backpropagation is then used to adjust the network weights so that in the following iteration the network makes its outputs closer to the ground truth. In this section, we discuss the main loss functions used to train deep video restoration methods.

3.3.1 Reconstruction loss

The most used loss function is the reconstruction loss, which measures the pixel-wise difference between restored and ground truth frames. Common reconstruction loss functions are L2 loss (Mean Squared Error) and L1 loss (Mean Absolute Error). L2 loss is known to have the problem of producing oversmooth results because of the low weight given to small errors. To alleviate this problem, several methods adopt loss functions based on L1 loss. Variants of simple L1 and L2 loss are Huber loss (Huber 1992), used to make the model less sensitive to outliers, and Charbonnier loss (Charbonnier et al. 1994), which adds a small term to be sure the loss will never be zero. The main drawback of using a reconstruction loss is that frames are compared without considering any kind of texture-awareness, which may lead to perceptually unsatisfying results. Therefore, using a reconstruction loss in combination with other types of loss functions is often preferred (Zhang et al. 2018a; Zhou et al. 2019; Li et al. 2020a; Chen et al. 2021; Paliwal et al. 2021).

3.3.2 Adversarial loss

In video restoration applying adversarial learning (Goodfellow et al. 2014) means using the restoration network as a generator and then adding a discriminator to judge whether the input frame is real or not. In this way, the generator can be improved by making

frames more and more similar to real ones, so that the discriminator will not be able to recognize them anymore. Since the task of the generator is more complex, the training typically starts from the generator, and the discriminator is added after a number of iterations (Lucas et al. 2019). The adversarial loss is useful to force the generator to remove some artifacts that may be still present in the restored frames. Paliwal et al. (2021) conditioned the discriminator using a gradient-based mask for the identification of textured regions, allowing it to detect high-frequency artifacts in smooth areas and classify them as fake, consequently encouraging the generator to remove them. In general, using only adversarial loss for training restoration methods leads to training instability (Gulrajani et al. 2017), and the restoration network may produce results substantially different from the desired ones (Mustafa et al. 2022). Consequently, the adversarial loss is often used in combination with the reconstruction loss, requiring a hyperparameter optimization for the regularization terms to weight the contribution of each loss (Zhang et al. 2018a; Paliwal et al. 2021).

3.3.3 Perceptual loss

The perceptual loss allows to assess the semantic difference between two frames and measures visual similarity by comparing frame content at feature level. The features are extracted by a neural network usually trained on other tasks, such as image classification. A common practice is to adopt VGG-based features (Chen and Koltun 2017) using a VGG model (Simonyan and Zisserman 2014). Although perceptual loss can produce perceptually satisfying results, using it alone may lead to training instability (Blau and Michaeli 2018). Therefore, it is usually used in combination with a reconstruction loss, with the additional cost of assigning the proper regularization term to each of the components of the total loss (Zhou et al. 2019). Using the perceptual loss adds a computational overhead to the training process, increasing the overall time required to train the network and the memory needed.

3.3.4 Temporal consistency loss

Temporal consistency is an important feature of video restoration methods because they should restore frames without introducing new temporal distortions, such as flickering. Although temporal consistency can be addressed by leveraging information from multiple frames, it can be further improved with the use of proper loss functions. A temporal consistency loss allows to enforce temporal coherence between consecutive frames by focusing on the temporal domain rather than on the spatial one. Typically, the output of the network at timestep t is compared to the outputs at timesteps $t-1$ and $t+1$, which are aligned with it via optical flow estimation. Different implementations of temporal consistency loss exist (Yue et al. 2020; Lai et al. 2018; Chen et al. 2021). Yue et al. (2020) first restored the frame at timestep t using its adjacent frames at timesteps $t-1$ and $t+1$, and then generated two new versions of the restored frame using two redundant noisy shots at timestep t, respectively. Finally, they imposed L1 loss between the restored frame and each of the two generated frames. Lai et al. (2018) proposed to employ a temporal consistency loss based on warping error between consecutive frames, that is, the output of the network at timestep $t-1$ is warped to the output at timestep t via optical flow estimation and L2 loss is computed between them. Similarly, Chen et al. (2021) used optical flow estimation to warp the previous restored frame to the current restored frame, and did the same for ground truth frames. Then, they computed L1 loss on the difference between

restored frames and the difference between ground truth frames. The application of temporal consistency loss is beneficial for video restoration methods because temporal consistency can be explicitly enforced via loss function and learned during training. When introduced into methods using a multi-frame baseline scheme, the main drawbacks of using a temporal consistency loss is that it requires either redundant computations (Yue et al. 2020) or a modification of the output of the network (Chen et al. 2021), as it requires to restore multiple frames in a single training iteration. It also requires to be used in combination with the reconstruction loss, requiring a proper regularization term (Yue et al. 2020; Chen et al. 2021). Moreover, optical flow computation in the temporal consistency loss increases the training time.

3.3.5 Detail-preserving loss

Restoration methods usually treat low and high frequencies in the same way, consequently producing oversmooth results (Hang et al. 2020). A detail-preserving loss allows restoration methods to improve their capability of recovering details by forcing the details contained within restored and ground truth frames to be the same. To this end, several solutions have been proposed (Li et al. 2020a; Xu et al. 2021; Isobe et al. 2020). Li et al. (2020a) used an edge detector to extract edge information from ground truth frames, generating a mask to highlight edges and force their model to pay more attention to them. Xu et al. (2021) introduced a loss function based on the Fast Fourier Transform (FFT) (Nussbaumer 1981): they computed the FFT on restored and ground truth frames and used L2 loss on both amplitude and phase components. Isobe et al. (2020) extracted high-frequency components on both restored and ground truth frames and computed a Charbonnier loss (Charbonnier et al. 1994) between them. Since the goal of a detail-preserving loss is to improve the detail recovery capability of neural networks, it should be used in combination with other loss functions, thus requiring a regularization term to weight its contribution in the overall loss (Li et al. 2020a; Xu et al. 2021; Isobe et al. 2020).

3.4 State of the art

Here we summarize the characteristics of the state-of-the-art video restoration methods introduced in the previous sections, according to the hierarchical organization in Fig. 3. Table 2 reports the main features of the architecture used (baseline scheme, design strategy, convolution type), how the methods handle motion (alignment technique and alignment level), and the loss functions used (reconstruction loss, adversarial loss, perceptual loss, temporal consistency loss, detail-preserving loss). For each method based on MEMC some details about how optical flow is computed and how the warping operation is performed are present. Besides, we report the number of frames used as input for those methods based on the multi-frame baseline scheme. Note that some methods in Table 2 have two baseline schemes, which means that they are recurrent methods but, at each timestep, they use a stack of frames as done by multi-frame methods.

4 Benchmark datasets

Video restoration methods based on deep learning require benchmark datasets both for training and evaluation. Through the years, several datasets have been proposed for the different restoration tasks. We summarize their characteristics in Table 3.

Table 2 Summary of the the state-of-the-art methods

Method name	Architecture			Motion handling		Loss function
	Baseline scheme	Design strategy	Convolution type	Alignment technique	Alignment level	
VESPCN Caballero et al. (2017)	Multi-frame (3)	Coarse-to-fine processing	2D	MEMC (STMC)	Frame	Reconstruction
DBN Su et al. (2017)	Multi-frame (5)	Residual learning (G)	2D	Implicit align	Feature	Reconstrution
STRCNN Hyun Kim et al. (2017)	Recurrent Multi-frame (5)	Residual learning (L)	2D	Implicit align	Feature	Reconstruction
DBLRGAN Zhang et al. (2018a)	Multi-frame (5)	Residual learning (L)	3D	Implicit align	Feature	Reconstruction Adversarial
DUF Jo et al. (2018)	Multi-frame (7)	Residual learning (G, L) Multi-path learning (G) Dense connections	2D 3D	Implicit align	Feature	Reconstruction
ToFlow Xue et al. (2019)	Multi-frame (7)	Residual learning (L)	2D	MEMC (SpyNet, STN)	Frame	Reconstruction
DVDNet Tassano et al. (2019)	Multi-frame (5)	Residual learning (G)	2D	MEMC (DeepFlow, BW)	Frame	Reconstruction
MFQE Yang et al. (2018)	Multi-frame (3)	Residual learning (G) Coarse-to-fine processing	2D	MEMC (STMC)	Frame	Reconstruction
MFQE2.0 Guan et al. (2019)	Multi-frame (3)	Residual learning (G, L) Dense connections Coarse-to-fine processing	2D	MEMC (STMC)	Frame	Reconstruction
STFAN Zhou et al. (2019)	Recurrent	Residual learning (G, L) Multi-path learning (G) Auto-regressive	2D	Implicit align	Feature	Reconstruction Perceptual
EDVR Wang et al. (2019)	Multi-frame (5)	Residual learning (G, L) Attention mechanism Coarse-to-fine processing	2D Deformable convs.	Deformable align	Feature	Reconstruction
ViDeNN Claus and Gemert (2019)	Multi-frame (3)	Residual learning (G)	2D	Implicit align	Feature	Reconstruction

Table 2 (continued)

Method name	Architecture			Motion handling		Loss function
	Baseline scheme	Design strategy	Convolution type	Alignment technique	Alignment level	
FITVNet Wang et al. (2020a)	Multi-frame (5)	Residual learning (G, L)	2D	Implicit align	Feature	Reconstruction
MB2D Park et al. (2020)	Multi-frame (3)	Coarse-to-fine processing	2D	Implicit align	Feature	Reconstruction
FastDVDNet Tassano et al. (2020)	Multi-frame (5)	Residual learning (G)	2D	Implicit align	Feature	Reconstruction
EVRNet Mehta et al. (2021)	Recurrent	Attention mechanism	2D Efficient (depth-wise)	Implicit align	Feature	Reconstruction
TDAN Tian et al. (2020b)	Multi-frame (5)	Residual learning (L)	2D Deformable convs.	Deformable align	Feature	Reconstruction
ESTRNN Zhong et al. (2020)	Recurrent	Residual learning (L) Dense connections Attention mechanism	2D	Implicit align	Feature	Reconstruction
STDF Deng et al. (2020)	Multi-frame (7)	Residual learning (G)	2D Deformable convs.	Deformable align	Frame	Reconstruction
VNLNet Davy et al. (2019)	Multi-frame (15)	Residual learning (G)	2D	Non-local search	Frame	Reconstruction
MuCAN Li et al. (2020a)	Multi-frame (5/7)	Residual learning (G, L) Attention mechanism	2D	Non-local search	Feature	Reconstruction Detail-preserving
RViDeNet Yue et al. (2020)	Multi-frame (3)	Residual learning (G, L) Attention mechanism	2D Deformable convs.	Deformable align	Feature	Reconstruction Temporal cons.
RSDN Isobe et al. (2020)	Recurrent	Residual learning (L) Multi-path learning (G)	2D	Implicit align	Feature	Reconstruction Detail-preserving
NL-ConvLSTM Xu et al. (2019)	Multi-frame (7)	Residual learning (G)	2D	Non-local search	Feature	Reconstruction
CDVD-TSP Pan et al. (2020)	Multi-frame (3)	Residual learning (G, L)	2D	MEMC (PWC-Net, BW)	Frame	Reconstruction

Table 2 (continued)

Method name	Architecture			Motion handling		Loss function
	Baseline scheme	Design strategy	Convolution type	Alignment technique	Alignment level	
PFNL Yi et al. (2019)	Multi-frame (5)	Residual learning (G, L)	2D	Non-local search	Frame	Reconstruction
MMNet Chen et al. (2021)	Multi-frame (7)	Residual learning (G) Multi-path learning (G)	2D 3D	Implicit align	Feature	Reconstruction Temporal cons.
BasicVSR Chan et al. (2021a)	Recurrent	Residual learning (L)	2D	MEMC (SpyNet, BW)	Feature	Reconstruction
BasicVSR++ Chan et al. (2022)	Recurrent	Residual learning (G, L)	2D Deformable convs.	MEMC (SpyNet, BW) Deformable align	Feature	Reconstruction
RFDA Zhao et al. (2021)	Recurrent Multi-frame (7)	Residual learning (G) Attention mechanism Multi-path learning (L)	2D Deformable convs.	Deformable align	Frame	Reconstruction
PVDNet Son et al. (2021)	Recurrent Multi-frame(3)	Residual learning (G, L)	2D	MEMC (LiteFlowNet, STN)	Frame	Reconstruction
MaskDNGAN Paliwal et al. (2021)	Multi-frame (5)	Residual learning (G, L) Attention mechanism	2D	MEMC (RAFT, BW)	Frame	Reconstruction Adversarial Perceptual
PaCNet Vaksman et al. (2021)	Multi-frame (7)	Residual learning (G)	2D 3D Efficient (separable)	Non-local search	Frame	Reconstruction
IFI-RNN Nah et al. (2019b)	Recurrent	Residual learning (L)	2D	Implicit align	Feature	Reconstruction

Following our hierarchical organization, we report the main characteristics of the architecture used, how it handles motion (alignment technique and alignment level), and the loss function used. Numbers in the baseline scheme column represent how many frames the methods require as input. G is for global, L is for local. BW means bilinear warping while STN indicates Spatial Transformer Networks

Table 3 Benchmark datasets for video restoration. DB, SR, DN, and CAR respectively mean deblurring, super-resolution, denoising, and compression artifact reduction

Dataset name	Task	Sequences				Frames				Frames per sequence		
		Train	Val	Test	Total	Train	Val	Test	Total	Average	Min	Max
GOPRO Nah et al. (2017)	DB	22	–	11	33	2103	–	1111	3214	97	48	150
DVD Su et al. (2017)	DB	61	–	10	71	5708	–	1000	6708	95	36	230
BSD Zhong et al. (2020)	DB	180	60	60	300	18,000	6000	6000	30,000	100	100	100
REDS Nah et al. (2019a)	DB SR	240	30	30	300	24,000	3000	3000	30,000	100	100	100
Vid4 Liu and Sun (2011)	SR	–	–	4	4	–	–	154	154	39	30	45
UDM10 Yi et al. (2019)	SR	–	–	10	10	–	–	320	320	32	32	32
SPMCS Tao et al. (2017)	SR	–	–	30	30	–	–	930	930	31	31	31
Vimeo90K Xue et al. (2019)	SR DN CAR	64,612	19,265	7824	91,701	452,284	134855	54,768	641,907	7	7	7
MFQEv2 Guan et al. (2019)	CAR	108	–	18	126	38,166	–	7980	46,146	366	51	2173
CRVD Yue et al. (2020)	DN	6	–	5	11	42	–	35	77	7	7	7
Set8 Tassano et al. (2019)	DN	–	–	8	8	–	–	2417	2417	302	29	690
DAVIS 2017 Pont-Tuset et al. (2017)	DN	60	30	30	120	4209	1999	2086	8294	69	25	127

Table 3 (continued)

Dataset name	Resolution	Format	Distortion type	Usage
GOPRO Nah et al. (2017)	1280 × 720	PNG	Synthetic	Train Test
DVD Su et al. (2017)	1280 × 720	JPEG	Synthetic	Train Test
BSD Zhong et al. (2020)	640 × 480	TIFF PNG	Real	Train Val Test
REDS Nah et al. (2019a)	1280 × 720	PNG JPEG	Synthetic	Train Val Test
Vid4 Liu and Sun (2011)	704× ∗	PNG	Synthetic	Test
UDM10 Yi et al. (2019)	1272 × 720	PNG	Synthetic	Test
SPMCS Tao et al. (2017)	960 × 540	PNG	Synthetic	Test
Vimeo90K Xue et al. (2019)	448 × 256	PNG	Synthetic	Train Val Test
MFQEv2 Guan et al. (2019)	*	YUV	Real	Train Test
CRVD Yue et al. (2020)	1920 × 1080	TIFF	Real	Train Test
Set8 Tassano et al. (2019)	960 × 540	PNG	Synthetic	Test
DAVIS 2017 Pont-Tuset et al. (2017)	∗ ×480	JPEG	Synthetic	Train Val Test

The symbol * means multiple sizes

Some datasets provide both degraded input and pristine ground truth sequences, while others only provide the pristine ground truth and the degraded input sequences must be synthetically generated. This solution could be feasible for video compression artifact reduction, because the artifacts introduced by using compression algorithms, such as JPEG2000 (Marcellin et al. 2000) or High Efficiency Video Coding (HEVC) (Sze et al. 2014), appear exactly as in the final application. Conversely, for video denoising, deblurring and super-resolution, this solution may not be optimal because the introduced distortions are merely an approximation of the real ones. For instance, artifacts introduced by adding Gaussian white noise are different from the ones derived from real low-light conditions.

Methods trained on synthetically generated approximated artifacts may perform suboptimally when applied to real-world distortions and, hence, creating datasets with realistic distortions is important to ensure the practical applicability of the restoration methods.

4.1 Datasets with real distortions

Creating video datasets containing real distortions, such as noise and blur, is a challenging task because this requires an acquisition system able to capture noisy/blurry and clean frames simultaneously. Different methods were proposed to generate paired datasets with videos affected by real-world artifacts. In the following, we shortly describe how existing datasets were created.

4.1.1 Beam-splitter deblurring (BSD) (Zhong et al. 2020)

The dataset was built using a beam splitter acquisition system with two synchronized cameras. The system could capture pairs of blurred and sharp videos in one shot by controlling the exposure time and the exposure intensity. A center-aligned synchronization scheme was adopted, so that the sharp exposure time lies exactly in the middle of the blurry exposure time. The dataset contains sharp/blurry videos captured at 15 frames per second (FPS) with different exposure times: 1ms-8ms, 2ms-16ms and 3ms-24ms.

4.1.2 Captured raw video denoising (CRVD) (Yue et al. 2020)

The dataset contains RAW videos captured using a surveillance camera at 20 FPS. Since capturing dynamic scenes using low International Organization for Standardization (ISO) generates motion blur, sequences containing objects were recorded, and the objects were manually moved to create object motion. For each static moment, multiple frames were captured, and the ground truth is obtained by averaging them, with the additional application of the Block-Matching and 3D filtering (BM3D) denoising algorithm (Dabov et al. 2007) to remove the remaining noise. Videos were captured using different ISO, ranging from 1600 to 25600, to capture different levels of noise.

4.1.3 MFQEv2 (Guan et al. 2019)

The dataset is composed of multiple sequences coming from different sources, i.e., Xiph.org[1], VQEG[2] and JCT-VC (Bossen 2013), containing different contents. The video sequences in this dataset are provided in the YUV domain without compression, and

[1] https://www.xiph.org/.

[2] https://www.its.bldrdoc.gov/vqeg/video-datasets-and-organizations.aspx.

compressed sequences are obtained using the HEVC compression standard (Sze et al. 2014). We called this dataset MFQEv2 to differentiate it from MFQE2.0, which is instead a state-of-the-art method.

4.2 Datasets with synthetic distortions

A common practice to generate video sequences for training video restoration methods is to take the clean video sequences and synthetically add the artifacts to obtain input/output pairs. Several datasets proposed for video restoration contain videos collected either from the web or from datasets for other related tasks, such as quality assessment or segmentation, that are synthetically distorted. In the following, we describe these datasets and the types of artifacts present.

4.2.1 GOPRO (Nah et al. 2017)

The dataset was generated using a camera capturing 240 FPS videos. Based on the idea that a long shutter speed can be approximated by averaging frames captured with a short shutter speed (i.e., 1/240 in the case of 240 FPS videos), each blurred frame is obtained by averaging from 7 to 13 sharp frames to produce different blur effects, and the mid-frame among the averaged frames is considered the ground truth.

4.2.2 Deep Video Deblurring (DVD) (Su et al. 2017)

Since a long exposure can be approximated by accumulating a number of short exposures (Telleen et al. 2007), motion blur at 30 FPS can be obtained by recording videos at 240 FPS, subsampling them every 8 frames and finally averaging each group of 7 consecutive frames. To use all the frames, optical flow was computed between adjacent high FPS frames to generate additional frames, which are then averaged. To avoid bias towards a specific device, different devices were used to capture the sequences. In addition, to avoid problems related to noise, all the sequences were recorded in good lighting conditions.

4.2.3 Realistic and Dynamic Scenes (REDS) (Nah et al. 2019a)

Proposed for the New Trends in Image Restoration and Enhancement (NTIRE) 2019 video restoration challenges, the dataset was recorded with a camera at 120 FPS. A CNN-based method (Niklaus et al. 2017) was used to increase the frame rate from 120 to 1920 FPS, and a duty cycle of 0.8 was used to generate blurry frames (from 1920 FPS sharp frames to 24 FPS blurry frames), whereas potential noise and compression artifacts were suppressed by downscaling the original frames. To better mimic the camera imaging pipeline and produce more realistic results, the Camera Response Function (CRF) and inverse CRF were estimated, and the blurry frames are computed in the signal space (obtained by applying the estimated inverse CRF) and converted back to the RGB color space (using the estimated CRF). For another challenge, additional distortions were introduced by compressing

the blurry frames using MPEG-4 (Sikora 1997) with quality 60%. Moreover, for video super-resolution, both the sharp and blurry frames were downscaled by a factor of four using bicubic interpolation.

4.2.4 Vimeo90K (Xue et al. 2019)

The dataset is composed of sequences with different contents downloaded from the Vimeo[3] video platform. Since only ground truth sequences are provided, any kind of artifact must be introduced synthetically. The authors of the dataset released the code to add noise, i.e., Gaussian noise and mixed noise (Gaussian + Salt & Pepper) for video denoising, to compress videos using the JPEG2000 algorithm (Marcellin et al. 2000) for video compression artifact reduction, and to reduce the spatial resolution by a factor of four using bicubic interpolation for video super-resolution.

4.2.5 Densely Annotated Video Segmentation 2017 (DAVIS) (Pont-Tuset et al. 2017)

Originally proposed for video object segmentation, this dataset is also employed in video restoration, in particular by video denoising methods. No code to add artifacts is provided.

5 Performance evaluation

5.1 Evaluation metrics

Defining common evaluation metrics to assess deep learning methods is important to objectively measure and compare their performance.

Metrics for the evaluation of restoration methods can be: (i) full-reference, which use reference frames; (ii) reduced-reference, which use partial information of reference frames (e.g., features); (iii) no-reference, which do not use any reference. Many metrics have been proposed to assess video quality (Li et al. 2019). Among them, the most common in video restoration are Peak Signal-to-Noise Ratio (PSNR) (Hore and Ziou 2010) and Structural Similarity Index (SSIM) (Wang et al. 2004). In the following, we describe them more in detail, and we mention other metrics seldom used.

5.1.1 Peak signal-to-noise ratio

Peak Signal-to-Noise Ratio (PSNR) (Hore and Ziou 2010) is a full-reference metric used to measure the quality of reconstruction algorithms. It is defined as the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. PSNR is based on a function of Mean Squared Error (MSE). When dealing with images, MSE allows to compare the true pixel values of the original image with those of the degraded one. Given two images I and K of size $n \times m$, where I is the original image and K is its degraded version, MSE is computed as follows:

[3] https://vimeo.com/.

$$MSE(I,K) = \frac{1}{n \times m} \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} (I_{i,j} - K_{i,j})^2 \tag{3}$$

Given MSE between I and K, PSNR is computed as follows:

$$PSNR(I,K) = 20 \cdot \log_{10} \frac{MAX}{\sqrt{MSE(I,K)}} \tag{4}$$

where MAX is the maximum pixel value of the dynamic range of the images, i.e., 255 for 8-bit images. Since MSE measures pixel errors, and low values of MSE imply high values of PSNR, the higher the PSNR, the better. When the compared images are identical, MSE is 0 and PSNR tends towards infinity.

5.1.2 Structural similarity index

Structural Similarity Index (SSIM) (Wang et al. 2004) is a full-reference metric for measuring the perceptual similarity between two images. SSIM considers image degradation as the perceived change in structural information, relying on the idea that image pixels have strong inter-dependencies, especially when they are spatially closed. These dependencies carry important information about the structure of the objects in the visual scene. Instead of using traditional error summation methods, such as PSNR, SSIM models image distortions as a combination of three factors: luminance distortion, contrast distortion and structural distortion. Given two images X and Y of the same size, SSIM is computed as follows:

$$SSIM(X,Y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{5}$$

where μ_x and μ_y are the average pixel values, σ_x^2 and σ_y^2 are the pixel variances and σ_{xy} is the pixel covariance of X and Y. The constants c_1 and c_2 are used to stabilize the division when the denominator is close to zero. They are respectively computed as $k_1 L$ and $k_2 L$, where L is the dynamic range of pixel values (255 for 8-bit images), $k_1 = 0.01$ and $k_2 = 0.03$ by default. SSIM assumes values in the [0, 1] range. Also in this case, the higher the SSIM, the better. When the compared images are identical, SSIM is equal to 1.

5.1.3 Other metrics

Zhang et al. (2018b) proposed Learned Perceptual Image Patch Similarity (LPIPS), a full-reference metric that first uses a pretrained CNN to extract neural features from both degraded and reference frames, and then compares them. MOtion-tuned Video Integrity Evaluation (MOVIE) index (Seshadrinathan and Bovik 2009) is a full-reference metric that uses a multi-scale framework to evaluate video fidelity, integrating both spatial and temporal aspects of distortion assessment. Soundararajan and Bovik (2012) proposed Spatio-Temporal Reduced Reference Entropic Differencies (STRRED), a reduced-reference metric that computes wavelet coefficients of frame differences modeled as Gaussian scale mixture, and measures the difference in the amount of spatial and temporal information contained in distorted and reference frames. Lai et al. (2018) proposed Warping Error (WE), a full-reference metric to evaluate temporal consistency of enhanced frames that makes use of optical flow to estimate pixel motion between two

Table 4 Performance of the state-of-the-art denoising methods

Method name	Dataset		Tested noise	PSNR	SSIM
	Train	Test			
ToFlow Xue et al. (2019)	Vimeo90K	Vimeo90K	AWGN ($\sigma = 15$)	36.63	0.963
			AWGN($\sigma = 25$)	34.89	0.952
EVRNet Mehta et al. (2021)	Vimeo90K	Vimeo90K	AWGN ($\sigma = 10$)	32.37	0.900
FITVNet Wang et al. (2020a)	Vimeo90K	Vimeo90K	AWGN ($\sigma = 15$)	37.70	0.955
			AWGN ($\sigma = 25$)	35.45	0.933
			AWGN ($\sigma = 35$)	33.89	0.912
			AWGN ($\sigma = 45$)	32.68	0.893
			AWGN($\sigma = 55$)	31.72	0.874
MMNet Chen et al. (2021)	Vimeo90K	Vimeo90K	AWGN ($\sigma = 10$)	41.26	0.978
			AWGN ($\sigma = 20$)	38.59	0.964
			AWGN ($\sigma = 30$)	36.85	0.950
			AWGN ($\sigma = 40$)	35.62	0.938
			AWGN ($\sigma = 50$)	34.60	0.927
	CRVD (sRGB)	CRVD (sRGB)	Real	38.79	0.978
RViDeNet Yue et al. (2020)	CRVD (RAW)	CRVD (RAW)	Real	43.97	0.987
	CRVD (sRGB)	CRVD (sRGB)	Real	39.95	0.979
MaskDNGAN Paliwal et al. (2021)	CRVD (RAW)	CRVD (RAW)	Real	43.96	0.988
	CRVD (sRGB)	CRVD (sRGB)	Real	40.40	0.981
DVDNet Tassano et al. (2019)	DAVIS 2017	DAVIS 2017	AWGN ($\sigma = 10$)	38.13	–
			AWGN ($\sigma = 20$)	35.70	–
			AWGN ($\sigma = 30$)	34.08	–
			AWGN ($\sigma = 40$)	32.86	–
			AWGN ($\sigma = 50$)	31.85	–
		Set8	AWGN ($\sigma = 10$)	36.08	–
			AWGN ($\sigma = 20$)	33.49	–
			AWGN ($\sigma = 30$)	31.79	–
			AWGN ($\sigma = 40$)	30.55	–
			AWGN ($\sigma = 50$)	29.56	–
FastDVDNet Tassano et al. (2020)	DAVIS 2017	DAVIS 2017	AWGN ($\sigma = 10$)	38.71	–
			AWGN ($\sigma = 20$)	35.77	–
			AWGN ($\sigma = 30$)	34.04	–
			AWGN ($\sigma = 40$)	32.82	–
			AWGN ($\sigma = 50$)	31.86	–
		Set8	AWGN ($\sigma = 10$)	36.44	–
			AWGN ($\sigma = 20$)	33.43	–
			AWGN($\sigma = 30$)	31.68	–
			AWGN ($\sigma = 40$)	30.46	–
			AWGN ($\sigma = 50$)	29.53	–
PaCNet Vaksman et al. (2021)	DAVIS 2017	DAVIS 2017	AWGN ($\sigma = 10$)	39.97	–
			AWGN ($\sigma = 20$)	36.82	–
			AWGN ($\sigma = 30$)	34.79	–
			AWGN ($\sigma = 40$)	33.34	–
			AWGN($\sigma = 50$)	32.20	–

Table 4 (continued)

Method name	Dataset		Tested noise	PSNR	SSIM
	Train	Test			
		Set8	AWGN ($\sigma = 10$)	37.06	–
			AWGN($\sigma = 20$)	33.94	–
			AWGN ($\sigma = 30$)	32.05	–
			AWGN ($\sigma = 40$)	30.70	–
			AWGN ($\sigma = 50$)	29.66	–

The tested noise column provides information about the type of noise considered. AWGN represents additive white Gaussian noise with standard deviation σ

Table 5 Performance of the state-of-the-art deblurring methods

Method name	Dataset		PSNR	SSIM
	Train	Test		
DBN Su et al. (2017)	DVD	DVD	30.05	0.964
STRCNN Hyun Kim et al. (2017)	–	DVD	29.11	–
MB2D Park et al. (2020)	GOPRO	GOPRO	32.16	0.953
	DVD	DVD	32.34	0.947
ESTRNN Zhong et al. (2020)	REDS	REDS	32.63	0.911
	DVD	DVD	31.07	0.902
	BSD (1–8 ms)	BSD (1–8 ms)	33.36	0.937
	BSD (2–16 ms)	BSD (2–16 ms)	31.95	0.925
	BSD (3–24 ms)	BSD (3–24 ms)	31.39	0.926
CDVD-TSP Pan et al. (2020)	GOPRO	GOPRO	31.67	0.928
	DVD	DVD	32.13	0.927
PVDNet Son et al. (2021)	DVD	DVD	32.31	0.926
	GOPRO	GOPRO	31.98	0.928
EDVR Wang et al. (2019)	REDS	REDS	36.96	0.966
DBLRGAN Zhang et al. (2018a)	DVD	DVD	33.19	–
STFAN Zhou et al. (2019)	DVD	DVD	31.24	0.934
IFI-RNN Nah et al. (2019b)	GOPRO	GOPRO	29.97	0.895
	DVD	DVD	30.80	0.899

adjacent frames, aligns them according to the estimated flow, and measures the pixel-wise error. Recently, Agarla et al. (2020, 2021) presented a no-reference video quality assessment method based on a CNN that approximates Mean Opinion Score (MOS) by considering both quality attributes, such as sharpness and noisiness, and semantics of videos.

Table 6 Performance of the state-of-the-art super-resolution methods

Method name	Dataset		Channel(s)	Degradation	PSNR	SSIM
	Train	Test				
VESPCN Caballero et al. (2017)	–	Vid4	Y	BI	25.35	0.756
DUF Jo et al. (2018)	–	Vid4	Y	BD	27.34	0.833
ToFlow Xue et al. (2019)	Vimeo90K	Vimeo90K	RGB	BI	33.08	0.942
		Vid4	RGB	BI	23.54	0.807
EDVR Wang et al. (2019)	REDS	REDS	RGB	BI	31.09	0.880
	Vimeo90K	Vimeo90K	RGB	BI	35.79	0.937
			Y		37.61	0.949
		Vid4	RGB	BI	25.83	0.808
			Y		27.35	0.826
EVRNet Mehta et al. (2021)	Vimeo90K	Vimeo90K	Y	BI	35.98	0.931
TDAN Tian et al. (2020b)	Vimeo90K	Vid4	RGB	BI	26.24	0.780
				BD	26.58	0.801
MuCAN Li et al. (2020a)	REDS	REDS	RGB	BI	30.88	0.875
	Vimeo90K	Vimeo90K	RGB	BI	35.49	0.934
			Y		37.32	0.947
RSDN Isobe et al. (2020)	Vimeo90K	Vimeo90K	RGB	BD	35.32	0.934
			Y		37.23	0.947
		Vid4	RGB	BD	26.43	0.835
			Y		27.92	0.851
		UDM10	RGB	BD	37.46	0.956
			Y		39.35	0.965
PFNL Yi et al. (2019)	–	Vid4	Y	BD	27.40	0.838
BasicVSR Chan et al. (2021a)	REDS	REDS	RGB	BI	31.42	0.891
	Vimeo90K	Vimeo90K	Y	BI	37.18	0.945
				BD	37.53	0.950
		Vid4	Y	BI	27.24	0.825
				BD	27.96	0.855
		UDM10	Y	BD	39.96	0.969
BasicVSR++ Chan et al. (2022)	REDS	REDS	RGB	BI	32.39	0.907
	Vimeo90K	Vimeo90K	Y	BI	37.79	0.950
				BD	38.21	0.955
		Vid4	Y	BI	27.79	0.840
				BD	29.04	0.875
		UDM10	Y	BD	40.72	0.972

Only the performance referring to ×4 upscaling factor is reported. In the degradation column, BI refers to bicubic downscaling, whereas BD to Gaussian downscaling. Y is the Y channel of the YCbCr color space

5.2 Performance evaluation of the methods

Here we analyze the performance of the state-of-the-art video restoration methods on the different restoration tasks. Tables 4, 5, 6 and 7 respectively report the performance of video denoising, video deblurring, video super-resolution and video compression artifact

Table 7 Performance of the state-of-the-art compression artifact reduction methods

Method name	Dataset		Tested compression	PSNR	SSIM	ΔPSNR	ΔSSIM
	Train	Test					
ToFlow Xue et al. (2019)	Vimeo90K	Vimeo90K	JPEG2000 ($q = 20$)	36.92	0.966	–	–
			JPEG2000 ($q = 40$)	34.97	0.953	–	–
			JPEG2000 ($q = 60$)	34.02	0.945	–	–
EVRNet Mehta et al. (2021)	Vimeo90K	Vimeo90K	JPEG2000 ($q = 20$)	36.65	0.967	–	–
			JPEG2000 ($q = 40$)	36.33	0.948	–	–
MFQE2.0 Guan et al. (2019)	MFQEv2	MFQEv2	HEVC ($QP = 42$)	–	–	0.59	1.65
			HEVC ($QP = 37$)	–	–	0.56	1.09
			HEVC ($QP = 32$)	–	–	0.52	0.68
			HEVC ($QP = 27$)	–	–	0.49	0.42
			HEVC ($QP = 22$)	–	–	0.46	0.27
STDF Deng et al. (2020)	MFQEv2	MFQEv2	HEVC ($QP = 37$)	–	–	0.83	1.51
			HEVC ($QP = 32$)	–	–	0.86	1.04
			HEVC ($QP = 27$)	–	–	0.72	0.57
			HEVC ($QP = 22$)	–	–	0.63	0.34
RFDA Zhao et al. (2021)	MFQEv2	MFQEv2	HEVC ($QP = 42$)	–	–	0.82	2.20
			HEVC ($QP = 37$)	–	–	0.91	1.62
			HEVC ($QP = 32$)	–	–	0.87	1.07
			HEVC ($QP = 27$)	–	–	0.82	0.68
			HEVC ($QP = 22$)	–	–	0.76	0.42

The tested compression column provides information about the compression algorithm used to compress videos. Performance on Vimeo90K is measured in RGB, while on MFQEv2 is measured on the Y channel of the YUV color space

reduction methods. For each method we report information about the datasets used for training and evaluation, and their performance in terms of PSNR and SSIM as reported in the original papers. We only considered the results obtained on the datasets reported in Sec. 4, even though some methods have also been evaluated on some less common or custom datasets (Pan et al. 2017; Maggioni et al. 2012). Note that entries in each table are grouped by method to highlight the source of the reported information. A direct comparison among different methods may be not fair since each of them is potentially trained with different settings (such as the software used for synthetic distortion generation).

Video denoising methods are commonly tested on videos containing additive white Gaussian noise (AWGN) (Xue et al. 2019; Mehta et al. 2021; Chen et al. 2021; Tassano et al. 2019, 2020; Wang et al. 2020a; Vaksman et al. 2021). Some methods (Paliwal et al. 2021; Yue et al. 2020; Chen et al. 2021) are also evaluated on real noisy scenes. Here, video denoising is performed either in the sRGB or in the RAW domain, directly processing the output of camera sensors. Based on the results reported in Table 4, MMNet (Chen et al. 2021) and PaCNet (Vaksman et al. 2021) are the best performing methods in removing AWGN from videos. Concerning the removal of real noise from sRGB frames, MaskDNGAN (Paliwal et al. 2021) can produce better results than

RViDeNet (Yue et al. 2020) and MMNet (Chen et al. 2021). In the RAW domain, MaskDNGAN (Paliwal et al. 2021) and RViDeNet (Yue et al. 2020) achieve almost the same denoising performance.

The performance of deblurring methods is reported in Table 5. Here the best methods are MB2D (Park et al. 2020) and PVDNet (Son et al. 2021). EDVR (Wang et al. 2019) and DLBRGAN (Zhang et al. 2018a) also achieve competitive performance.

Video super-resolution can be performed using different upscaling factors, i.e., ×2, ×3 and ×4. In Table 6 we only report the performance obtained by video super-resolution methods using the ×4 upscaling factor, which is the most common one. Two degradation types are usually evaluated: bicubic downscaling (BI), which is performed by downscaling frames using bicubic interpolation, and Gaussian downscaling (BD), which is performed by applying a Gaussian filter (with standard deviation $\sigma = 1.6$) to frames and then downscaling them using bicubic interpolation. The performance on the Y channel of the YCbCr color space is usually evaluated in addition to the one in RGB. BasicVSR++ (Chan et al. 2022) achieves the best performance on all the considered datasets, color channels, and degradation types, demonstrating its superiority compared to the other methods. It is followed by BasicVSR (Chan et al. 2021a) and EDVR (Wang et al. 2019), which obtain competitive performance.

Table 7 documents the results obtained by video compression artifact reduction methods in restoring videos compressed using JPEG2000 (Marcellin et al. 2000) and HEVC (Sze et al. 2014). ToFlow (Xue et al. 2019) achieves PSNR higher than EVRNet (Mehta et al. 2021) in removing compression artifacts introduced by JPEG2000 when the compression is high ($q = 20$), while the latter is considerably better when the compression is lower (q is higher). The two methods are equal in terms of SSIM. The performance on MFQEv2 (Guan et al. 2019) is commonly measured using ΔPSNR and ΔSSIM: ΔPSNR is obtained as PSNR($\hat{F}, \bar{F}$) - PSNR($F, \bar{F}$), where $\hat{F}$ is the enhanced frame, $\bar{F}$ is the ground truth frame and F is the compressed frame; ΔSSIM is computed in a similar way. The higher the ΔPSNR and ΔSSIM, the better. Moreover, the restoration performance is evaluated on the Y channel of the YUV color space. The best performing method is RFDA (Zhao et al. 2021), which obtains the highest ΔPSNR and ΔSSIM at every compression level. It is followed by STDF (Deng et al. 2020) and MFQE2.0 (Guan et al. 2019).

Efficiency is another important criterion for the evaluation of video restoration methods. In Table 8, we report the results using five metrics commonly adopted to evaluate efficiency of CNNs. Giga operations per second (GOPs), Giga floating point operations per second (GFLOPs) and Giga multiply-accumulate operations per second (GMACs) refer to the number of operations performed in one second. The lower, the better. Runtime reports how many seconds the methods require to restore a frame at a given resolution. All the values are taken from the original papers. Note that the methods may not be directly comparable because these metrics were computed on different devices and using different software, which might produce slightly different results. In addition, runtime is computed using different Graphic Processor Units (GPUs), whose performance change based on the specific model. For video super-resolution methods we report only information using the ×4 upscaling factor. Since the number of operations performed by the methods is positively correlated with the running time, i.e., a higher number of operations implies a higher running time (Bianco et al. 2018), here we comment only aspects related to the running time.

We take into account high-resolution videos, i.e., videos containing frames whose size is greater than 1280×720 pixels. Since real videos at 30 FPS require a processing time lower than 0.03 seconds for each frame, we can observe that none of these methods can achieve real-time restoration performance even using high-performing GPUs. Based on the

Table 8 Efficiency of the state-of-the-art video restoration methods

Method name	Parameters	GOPs	GFLOPs	GMACs	Runtime	Device
VESPCN Caballero et al. (2017)	–	14 @ 1920 × 1080	–	–	–	GRID K2
DBN Su et al. (2017)	–	–	–	–	> 1s @ 1280 × 720	Titan X
STRCNN Hyun Kim et al. (2017)	–	–	–	–	0.13s @ 1280 × 720 0.042s @ 640 × 480	GTX 1080
DUF Jo et al. (2018)	–	–	–	–	2.82s @ 1920 × 1080	GTX 1080 Ti
ToFlow Xue et al. (2019)	–	–	–	–	0.4s @ 448 × 256	Titan X
DVDNet Tassano et al. (2019)	–	–	–	–	8s @ 960 × 540	Titan Xp
MFQE2.0 Guan et al. (2019)	255K	–	–	–	0.62s @ 1920 × 1080 0.27s @ 1280 × 720 0.12s @ 832 × 480 0.04s @ 416 × 240	GTX 1080 Ti
STFAN Zhou et al. (2019)	5.37M	–	–	–	0.15s @ 1280 × 720	Titan Xp
EDVR Wang et al. (2019)	–	–	936.5 @ 1280 × 720	–	–	Titan Xp
ViDeNN Claus and Gemert (2019)	–	–	–	–	0.33s @ 1920 × 1080 0.07s @ 352 × 288	Titan X
FITVNet Wang et al. (2020a)	–	–	–	–	0.047s @ 448 × 256	RTX 2080 Ti
MB2D Park et al. (2020)	5.42M	–	–	–	0.27s @ 1280 × 720	Titan V
FastDVDNet Tassano et al. (2020)	–	–	–	–	0.1s @ 960 × 540	Titan Xp
EVRNet Mehta et al. (2021)	79.55K 78.71K	–	–	10.39 @ 448 × 256 10.13 @ 448 × 256	–	–

Table 8 (continued)

Method name	Parameters	GOPs	GFLOPs	GMACs	Runtime	Device
TDAN Tian et al. (2020b)	1.97M	–	–	–	–	GTX 1080 Ti
ESTRNN Zhong et al. (2020)	–	–	–	206.70 @ 1280×720	–	–
STDF Deng et al. (2020)	1.2M	–	204.08 @ 832×480	–	1s @ 1920×1080 0.4s @ 1280×720 0.169s @ 832×480 0.042s @ 416×240	GTX 1080 Ti
RViDeNet Yue et al. (2020)	–	–	–	–	–	RTX 2080 Ti
RSDN Isobe et al. (2020)	6.19M	–	350 @ 1280×720 130 @ 720×480 40 @ 448×256	–	0.094s @ 1280×720 0.015s @ 448×256	Tesla V100
CDVD-TSP Pan et al. (2020)	16.19M	–	–	–	–	–
PFNL Yi et al. (2019)	4.14M	–	–	–	0.41s @ 1920×1080	GTX 1080 Ti
MMNet Chen et al. (2021)	–	–	–	–	0.003s @ 960×540 0.001s @ 448×256	Titan RTX
BasicVSR Chan et al. (2021a)	6.3M	–	–	–	0.063s @ 1280×720	Tesla V100
BasicVSR++ Chan et al. (2022)	7.3M	–	–	–	0.077s @ 1280×720	–
RFDA Zhao et al. (2021)	840K	–	–	–	0.274s @ 1920×1080 0.123s @ 1280×720 0.056s @ 640×480	RTX 3090
PVDNet Son et al. (2021)	10.5M	–	–	936 @ 1280×720	0.11s @ 1280×720	Titan Xp

Table 8 (continued)

Method name	Parameters	GOPs	GFLOPs	GMACs	Runtime	Device
MaskDNGan Paliwal et al. (2021)	–	–	–	–	1.47s @ 1920×1080	RTX 2080 Ti
PaCNet Vaksman et al. (2021)	2.87M	–	–	–	30s @ 854×480	Quadro RTX 8000
IFI-RNN Nah et al. (2019b)	–	–	–	–	0.035s @ 960×540	GTX 1080 Ti

For EVRNet, the first row refers to super-resolution, while the second one to denoising and compression artifact reduction

For video super-resolution methods only the performance referring to $\times 4$ upscaling factor is reported

results in Table 8, RSDN (Isobe et al. 2020) and BasicVSR (Chan et al. 2021a) are the most efficient methods, approaching real-time restoration performance.

EVRNet (Mehta et al. 2021) uses two slight different models to perform the different tasks: one for super-resolution (first row) and one for denoising and compression artifact reduction (second row). In contrast to the other models, it is very lightweight because it was designed to work on edge devices, such as smartphones. PaCNet (Vaksman et al. 2021) requires about 30 seconds to restore a frame at 854×480 resolution even if it has a limited amount of parameters. This is due to the preliminary alignment process based on non-local search that explicitly tries to craft artificial frames by aggregating similar patches coming from adjacent frames. DVDNet (Tassano et al. 2019) requires about 8 seconds on frames at 960×540 resolution, where 6 seconds are dedicated to the alignment process performed using MEMC.

6 Challenges and future trends

Despite the progress made in video restoration using deep learning, there are still many issues to address. In this section, we point out the main challenges and future trends as emerged from the analysis presented in this paper.

6.1 Real-time restoration

State-of-the-art video restoration methods are characterized by high reconstruction performance. Nevertheless, efficiency still represents an obstacle that makes their application to several real-world problems challenging, especially those requiring real-time computations. Recent methods are typically evaluated on highly performing hardware, such as GPUs, that may not be available in some practical scenarios. Due to the increasing popularity of mobile devices, for example, one may expect to run these models on smartphones and hand-held cameras, which are characterized by limited resources in terms of computational power, memory, and battery consumption. Designing lightweight models able to run on such devices in real time would considerably extend their applicability to real-world problems, and investigations towards this direction are important.

6.2 Improved alignment strategies

The effectiveness of video restoration methods strictly depends on the adopted solution for motion handling. Methods based on optical flow are sensitive to light changes, fast motion, and occluded objects, while methods using implicit alignment are limited by the local receptive field of standard convolutions. Some solutions, such as deformable convolutions, were proposed to address these limitations, but they introduce training instability and increase computational complexity. According to the investigation made by Chan et al. (2021b, 2022), a possible future trend is the exploration of the relationships among existing alignment strategies, with the purpose of developing new solutions that combine all the underlying advantages.

6.3 All-in-one video restoration methods

Most of the video restoration methods proposed during the past few years tackle only one restoration task. Although some methods demonstrated to be flexible to different types of distortion (Xue et al. 2019; Wang et al. 2019; Mehta et al. 2021), they have been optimized for only one task at a time. In real-world scenarios, videos may be simultaneously affected by multiple distortions, because artifacts are introduced at different levels of the camera pipeline: for example, noisy videos are also later compressed. Therefore, designing robust all-in-one methods that can address multiple restoration tasks at the same time, i.e., restoring videos containing multiple distortion types, would extend their applicability to real-world cases. Some methods towards this direction have been recently developed (Rota et al. 2022; Katsaros et al. 2021).

6.4 More representative evaluation metrics

Common metrics for the evaluation of video restoration methods are PSNR and SSIM. However, their values are not well correlated to human perception, meaning that high values of these metrics can be obtained even if the results are unpleasant for humans. To this end, several metrics that better correlate to human perception have been proposed, both for image (Zhang et al. 2018b; Kim and Lee 2017; Reisenhofer et al. 2018) and video assessment (Park et al. 2012; Bampis et al. 2018; Agarla et al. 2020, 2021), but currently there is not a globally-accepted measure for video restoration. Thus, there is the need to define and converge to an accurate and perceptual-based metric for the evaluation of restoration results. Temporal consistency is an important aspect of video restoration, but it is usually underestimated and only occasionally evaluated. In most video restoration papers only metrics applied to each individual frame are typically used, without taking into account any dependency among them. It would be instead appropriate to employ metrics also for temporal consistency evaluation, such as STRRED (Soundararajan and Bovik 2012), MOVIE (Seshadrinathan and Bovik 2009) or Warping Error (Lai et al. 2018).

6.5 Datasets with realistic distortions

Despite the large availability of video datasets for training video restoration methods, the distortions they contain are usually synthetically generated (e.g., noise is typically modeled as additive Gaussian white noise and downscaling degradation is modeled using interpolation methods). Since real-world distortions could have different characteristics with respect to synthetic ones, methods trained on these datasets may underperform when applied to real scenarios. Some datasets with realistic artifacts were proposed (Zhong et al. 2020; Yue et al. 2020), but the difficulty of the collection task largely constrained the acquisition conditions, thereby limiting their potential applicability. Developing complex acquisition systems able to model realistic distortions is a challenge, but could be beneficial to extend the applicability of restoration methods to real-world tasks.

6.6 Combining traditional and deep learning methods

Video restoration methods based on deep learning have three main disadvantages with respect to traditional methods (López-Tapia et al. 2021): (i) they are less frequently found to incorporate domain knowledge, which in turn makes them less robust to videos containing unseen degradations; (ii) they need a large amount of data to learn the non-linear mapping between inputs and outputs, which requires a time-consuming video collection process; (iii) they are less interpretable, which limits their applicability to some sensitive contexts. These problems could potentially be tackled using Deep Unfolding Networks (DUNs), which implement the conventional iterative optimization process of traditional methods using deep neural networks (Gregor and LeCun 2010). Despite many works adopting DUNs have been proposed for different image restoration tasks (Dong et al. 2018; Zhang et al. 2020; Gong et al. 2020; Li et al. 2020b; Ren et al. 2021), fewer are designed for the video domain (Chiche et al. 2020; Sun et al. 2021).

7 Conclusions

In this paper, we provided a review of video restoration methods based on deep learning. We selected well-established and recent methods for video restoration, and analyzed in a structured manner their main features related to architectural choices, strategies for motion handling, and loss functions.

For each restoration task we detailed the characteristics of benchmark datasets and classified them based on the types of distortions they contain. Despite the large availability of video datasets, we highlighted that most of them contain synthetic distortions that may differ from real ones, limiting the applicability of video restoration methods.

The main evaluation criteria are also discussed and used to compare the performance of the considered methods, providing an overview the most promising methods in terms of both effectiveness and efficiency. We noticed that even if video restoration quality made much progress in recent years, video restoration methods cannot yet restore high-resolution frames in real time.

Possible improvements of the research include the development of methods able to run on resource-limited devices in real time, the study of more robust alignment strategies, the development of methods to address multiple restoration tasks at the same time, the definition of more suitable and globally-accepted metrics for result evaluation, the acquisition of freely available datasets containing real-world distortions, and the combination of traditional and deep learning methods.

Author contributions All authors contributed equally to this work.

Funding Open access funding provided by Università degli Studi di Milano - Bicocca within the CRUI-CARE Agreement. No funding was received for conducting this study.

Data availability The datasets generated during and/or analyzed during the current study are available in the following repositories: GOPRO (https://seungjunnah.github.io/Datasets/gopro) DVD (https://www.cs.ubc.ca/labs/imager/tr/2017/DeepVideoDeblurring/) BSD (https://github.com/zzh-tech/ESTRNN) REDS (https://seungjunnah.github.io/Datasets/reds.html) Vid4(https://people.csail.mit.edu/celiu/CVPR2011/default.html) UDM10 (https://github.com/psychopa4/PFNL) SPMCS (https://github.com/jiangsutx/SPMC_VideoSR) Vimeo90K (http://toflow.csail.mit.edu/) MFQEv2 (https://github.com/RyanXingQL/MFQEv2.0)

CRVD (https://github.com/cao-cong/RViDeNet) Set8 (https://github.com/m-tassano/fastdvdnet) Davis2017 (https://davischallenge.org/davis2017/code.html)

Code availability Not applicable.

Declarations

Competing interest The authors have no competing interests to declare that are relevant to the content of this article.

Ethical approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

References

Agarla M, Celona L, Schettini R (2020) No-reference quality assessment of in-capture distorted videos. J Imaging 6(8):74

Agarla M, Celona L, Schettini R (2021) An efficient method for no-reference video quality assessment. J Imaging 7(3):55

Amiaz T, Lubetzky E, Kiryati N (2007) Coarse to over-fine optical flow estimation. Pattern Recogn 40(9):2496–2503

Bampis CG, Li Z, Bovik AC (2018) Spatiotemporal feature integration and model fusion for full reference video quality assessment. IEEE Trans Circ Syst Video Technol 29(8):2256–2270

Bao G, Graeber MB, Wang X (2020) Depthwise multiception convolution for reducing network parameters without sacrificing accuracy. In: 16th international conference on control, automation, robotics and vision (ICARCV). IEEE, pp 747–752

Beauchemin SS, Barron JL (1995) The computation of optical flow. ACM Comput Surv 27(3):433–466

Bianco S, Cadene R, Celona L et al (2018) Benchmark analysis of representative deep neural network architectures. IEEE Access 6:64270–64277

Blau Y, Michaeli T (2018) The perception-distortion tradeoff. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6228–6237

Bossen F (2013) Common test conditions and software reference configurations. JCTVC-L1100 12(7)

Bovik AC (2009) The essential guide to video processing. Academic Press, New York

Caballero J, Ledig C, Aitken A, et al (2017) Real-time video super-resolution with spatio-temporal networks and motion compensation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4778–4787

Chan KC, Wang X, Yu K et al (2021a) Basicvsr: the search for essential components in video super-resolution and beyond. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 4947–4956

Chan KC, Wang X, Yu K et al (2021b) Understanding deformable alignment in video super-resolution. In: AAAI conference on artificial intelligence

Chan KC, Zhou S, Xu X et al (2022) Basicvsr++: improving video super-resolution with enhanced propagation and alignment. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 5972–5981

Charbonnier P, Blanc-Feraud L, Aubert G et al (1994) Two deterministic half-quadratic regularization algorithms for computed imaging. In: Proceedings of 1st international conference on image processing. IEEE, pp 168–172
Chen Q, Koltun V (2017) Photographic image synthesis with cascaded refinement networks. In: Proceedings of the IEEE international conference on computer vision, pp 1511–1520
Chen H, Jin Y, Xu K et al (2021) Multiframe-to-multiframe network for video denoising. IEEE Trans Multimed 24:2164–2178
Chiche BN, Frontera-Pons J, Woiselle A et al (2020) Deep unrolled network for video super-resolution. In: 2020 Tenth international conference on image processing theory, tools and applications (IPTA). IEEE, pp 1–6
Chollet F (2017) Xception: deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1251–1258
Claus M, van Gemert J (2019) Videnn: deep blind video denoising. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops
Dabov K, Foi A, Katkovnik V et al (2007) Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE Trans Image Process 16(8):2080–2095
Dai J, Qi H, Xiong Y et al (2017) Deformable convolutional networks. In: Proceedings of the IEEE international conference on computer vision, pp 764–773
Davy A, Ehret T, Morel JM et al (2019) A non-local cnn for video denoising. In: 2019 IEEE international conference on image processing (ICIP). IEEE, pp 2409–2413
Deng J, Wang L, Pu S et al (2020) Spatio-temporal deformable convolution for compressed video quality enhancement. In: Proceedings of the AAAI conference on artificial intelligence, pp 10696–10703
Dong W, Wang P, Yin W et al (2018) Denoising prior driven deep neural network for image restoration. IEEE Trans Pattern Anal Mach Intell 41(10):2305–2318
Dosovitskiy A, Fischer P, Ilg E et al (2015) Flownet: learning optical flow with convolutional networks. In: Proceedings of the IEEE international conference on computer vision, pp 2758–2766
Fan Y, Yu J, Liu D et al (2019) An empirical investigation of efficient spatio-temporal modeling in video restoration. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops
Gong D, Zhang Z, Shi Q et al (2020) Learning deep gradient descent optimization for image deconvolution. IEEE Trans Neural Netw Learn Syst 31(12):5468–5482
Goodfellow I, Pouget-Abadie J, Mirza M et al (2014) Generative adversarial nets. Adv Neural Inf Process Syst 27
Gregor K, LeCun Y (2010) Learning fast approximations of sparse coding. In: Proceedings of the 27th international conference on international conference on machine learning, pp 399–406
Guan Z, Xing Q, Xu M et al (2019) Mfqe 2.0: a new approach for multi-frame quality enhancement on compressed video. IEEE Trans Pattern Anal Mach Intell 43(3):949–963
Gulrajani I, Ahmed F, Arjovsky M et al (2017) Improved training of wasserstein gans. Adv Neural Inf Process Syst 30
Guo MH, Xu TX, Liu JJ et al (2022) Attention mechanisms in computer vision: a survey. Comput Visual Med 1–38
Hang Y, Liao Q, Yang W et al (2020) Attention cube network for image restoration. In: Proceedings of the 28th ACM international conference on multimedia, pp 2562–2570
Haris M, Shakhnarovich G, Ukita N (2019) Recurrent back-projection network for video super-resolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 3897–3906
He K, Zhang X, Ren S et al (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
Hore A, Ziou D (2010) Image quality metrics: Psnr vs. ssim. In: 2010 20th international conference on pattern recognition. IEEE, pp 2366–2369
Howard AG, Zhu M, Chen B et al (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861
Hu Y, Li Y, Song R (2017) Robust interpolation of correspondences for large displacement optical flow. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 481–489
Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7132–7141
Huang G, Liu Z, Van Der Maaten L et al (2017a) Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4700–4708
Huang Y, Wang W, Wang L (2017b) Video super-resolution via bidirectional recurrent convolutional networks. IEEE Trans Pattern Anal Mach Intell 40(4):1015–1028

Huber PJ (1992) Robust estimation of a location parameter. In: Breakthroughs in statistics. Springer, pp 492–518
Hui TW, Tang X, Loy CC (2018) Liteflownet: a lightweight convolutional neural network for optical flow estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 8981–8989
Hyun Kim T, Mu Lee K, Scholkopf B et al (2017) Online video deblurring via dynamic temporal blending network. In: Proceedings of the IEEE international conference on computer vision, pp 4038–4047
Isobe T, Jia X, Gu S et al (2020) Video super-resolution with recurrent structure-detail network. In: European conference on computer vision. Springer, pp 645–660
Jaderberg M, Simonyan K, Zisserman A et al (2015) Spatial transformer networks. Adv Neural Inf Process Syst 28:2017–2025
Jia X, De Brabandere B, Tuytelaars T et al (2016) Dynamic filter networks. Adv Neural Inf Process Syst 29:667–675
Jo Y, Oh SW, Kang J et al (2018) 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, pp 3224–3232
Katsaros E, Ostrowski PK, Wesierski D et al (2021) Concurrent video denoising and deblurring for dynamic scenes. IEEE Access 9:157437–157446
Kim J, Lee S (2017) Deep learning of human visual sensitivity in image quality assessment framework. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1676–1684
Koh J, Lee J, Yoon S (2021) Single-image deblurring with neural networks: a comparative survey. Comput Vis Image Underst 203(103):134
Lai WS, Huang JB, Wang O et al (2018) Learning blind video temporal consistency. In: Proceedings of the European conference on computer vision (ECCV), pp 170–185
Li D, Jiang T, Jiang M (2019) Recent advances and challenges in video quality assessment. ZTE Commun 17(1):3–11
Li W, Tao X, Guo T et al (2020a) Mucan: multi-correspondence aggregation network for video super-resolution. In: European conference on computer vision, Springer, pp 335–351
Li Y, Tofighi M, Geng J et al (2020b) Efficient and interpretable deep blind image deblurring via algorithm unrolling. IEEE Trans Comput Imaging 6:666–681
Lim B, Son S, Kim H et al (2017) Enhanced deep residual networks for single image super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 136–144
Liu C, Sun D (2011) A Bayesian approach to adaptive video super resolution. In: CVPR 2011. IEEE, pp 209–216
Liu H, Ruan Z, Zhao P et al (2022) Video super-resolution based on deep learning: a comprehensive survey. Artif Intell Rev 1–55
Liu J, Liu D, Yang W et al (2020) A comprehensive benchmark for single image compression artifact reduction. IEEE Trans Image Process 29:7845–7860
López-Tapia S, Molina R, Katsaggelos AK (2021) Deep learning approaches to inverse problems in imaging: Past, present and future. Digital Signal Process 119(103):285
Lucas A, Lopez-Tapia S, Molina R et al (2019) Generative adversarial networks and perceptual losses for video super-resolution. IEEE Trans Image Process 28(7):3312–3327
Maggioni M, Boracchi G, Foi A et al (2012) Video denoising, deblocking, and enhancement through separable 4-d nonlocal spatiotemporal transforms. IEEE Trans Image Process 21(9):3952–3966
Marcellin MW, Gormish MJ, Bilgin A et al (2000) An overview of jpeg-2000. In: Proceedings DCC 2000. Data compression conference. IEEE, pp 523–541
Mehta S, Kumar A, Reda F et al (2021) Evrnet: efficient video restoration on edge devices. In: Proceedings of the 29th ACM international conference on multimedia, pp 983–992
Mustafa A, Mikhailiuk A, Iliescu DA et al (2022) Training a task-specific image reconstruction loss. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision, pp 2319–2328
Nah S, Hyun Kim T, Mu Lee K (2017) Deep multi-scale convolutional neural network for dynamic scene deblurring. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3883–3891
Nah S, Baik S, Hong S et al (2019a) Ntire 2019 challenge on video deblurring and super-resolution: dataset and study. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops

Nah S, Son S, Lee KM (2019b) Recurrent neural networks with intra-frame iterations for video deblurring. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 8102–8111
Niklaus S, Mai L, Liu F (2017) Video frame interpolation via adaptive separable convolution. In: Proceedings of the IEEE international conference on computer vision, pp 261–270
Nussbaumer HJ (1981) The fast fourier transform. In: Fast fourier transform and convolution algorithms. Springer, pp 80–111
Paliwal A, Zeng L, Kalantari NK (2021) Multi-stage raw video denoising with adversarial loss and gradient mask. In: 2021 IEEE international conference on computational photography (ICCP). IEEE, pp 1–10
Pan L, Dai Y, Liu M et al (2017) Simultaneous stereo video deblurring and scene flow estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4382–4391
Pan J, Bai H, Tang J (2020) Cascaded deep video deblurring using temporal sharpness prior. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 3043–3051
Park J, Seshadrinathan K, Lee S et al (2012) Video quality pooling adaptive to perceptual distortion severity. IEEE Trans Image Process 22(2):610–620
Park D, Kang DU, Chun SY (2020) Blur more to deblur better: multi-blur2deblur for efficient video deblurring. arXiv preprint arXiv:2012.12507
Pont-Tuset J, Perazzi F, Caelles S et al (2017) The 2017 davis challenge on video object segmentation. arXiv preprint arXiv:1704.00675
Ranjan A, Black MJ (2017) Optical flow estimation using a spatial pyramid network. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4161–4170
Reisenhofer R, Bosse S, Kutyniok G et al (2018) A Haar wavelet-based perceptual similarity index for image quality assessment. Signal Process 61:33–43
Ren C, He X, Wang C et al (2021) Adaptive consistency prior based deep network for image denoising. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 8596–8606
Revaud J, Weinzaepfel P, Harchaoui Z et al (2015) Epicflow: edge-preserving interpolation of correspondences for optical flow. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1164–1172
Rota C, Buzzelli M (2022) Mdvrnet: deep video restoration under multiple distortions. In: Proceedings of the 17th international joint conference on computer vision, imaging and computer graphics theory and applications, vol 4. VISAPP, pp 419–426
Savian S, Elahi M, Tillo T (2020) Optical flow estimation with deep learning, a survey on recent advances. In: Deep biometrics. Springer, pp 257–287
Seshadrinathan K, Bovik AC (2009) Motion tuned spatio-temporal quality assessment of natural videos. IEEE Trans Image Process 19(2):335–350
Sikora T (1997) The mpeg-4 video standard verification model. IEEE Trans Circ Syst Video Technol 7(1):19–31
Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
Son H, Lee J, Lee J et al (2021) Recurrent video deblurring with blur-invariant motion estimation and pixel volumes. ACM Trans Graphics 40(5):1–18
Soundararajan R, Bovik AC (2012) Video quality assessment by reduced reference spatio-temporal entropic differencing. IEEE Trans Circ Syst Video Technol 23(4):684–694
Su S, Delbracio M, Wang J et al (2017) Deep video deblurring for hand-held cameras. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1279–1288
Sun D, Yang X, Liu MY et al (2018) Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 8934–8943
Sun L, Dong W, Li X et al (2021) Deep maximum a posterior estimator for video denoising. Int J Comput Vis 129(10):2827–2845
Sze V, Budagavi M, Sullivan GJ (2014) High efficiency video coding (hevc). In: Integrated circuit and systems, algorithms and architectures, vol 39. Springer, p 40
Szegedy C, Liu W, Jia Y et al (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9
Tao X, Gao H, Liao R et al (2017) Detail-revealing deep video super-resolution. In: Proceedings of the IEEE international conference on computer vision, pp 4472–4480
Tassano M, Delon J, Veit T (2019) Dvdnet: a fast network for deep video denoising. In: 2019 IEEE international conference on image processing (ICIP). IEEE, pp 1805–1809

Tassano M, Delon J, Veit T (2020) Fastdvdnet: towards real-time deep video denoising without flow estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 1354–1363
Teed Z, Deng J (2020) Raft: recurrent all-pairs field transforms for optical flow. In: European conference on computer vision. Springer, pp 402–419
Telleen J, Sullivan A, Yee J et al (2007) Synthetic shutter speed imaging. In: Computer graphics forum. Wiley, New York, pp 591–598
Tian C, Fei L, Zheng W et al (2020a) Deep learning on image denoising: an overview. Neural Netw 131:251–275
Tian Y, Zhang Y, Fu Y et al (2020b) Tdan: temporally-deformable alignment network for video super-resolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 3360–3369
Tran D, Bourdev L, Fergus R et al (2015) Learning spatiotemporal features with 3d convolutional networks. In: Proceedings of the IEEE international conference on computer vision, pp 4489–4497
Vaksman G, Elad M, Milanfar P (2021) Patch craft: video denoising by deep modeling and patch matching. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV), pp 2157–2166
Wang Z, Bovik AC, Sheikh HR et al (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13(4):600–612
Wang X, Girshick R, Gupta A et al (2018) Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7794–7803
Wang X, Chan KC, Yu K et al (2019) Edvr: video restoration with enhanced deformable convolutional networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops
Wang C, Zhou SK, Cheng Z (2020a) First image then video: a two-stage network for spatiotemporal video denoising. arXiv preprint arXiv:2001.00346
Wang Z, Chen J, Hoi SC (2020b) Deep learning for image super-resolution: a survey. IEEE Trans Pattern Anal Mach Intell 43(10):3365–3387
Weinzaepfel P, Revaud J, Harchaoui Z et al (2013) Deepflow: Large displacement optical flow with deep matching. In: Proceedings of the IEEE international conference on computer vision, pp 1385–1392
Woo S, Park J, Lee JY et al (2018) Cbam: convolutional block attention module. In: Proceedings of the european conference on computer vision (ECCV), pp 3–19
Xiao Z, Zhang Z, Hung KW et al (2021) Real-time video super-resolution using lightweight depthwise separable group convolutions with channel shuffling. J Vis Commun Image Represent 75(103):038
Xingjian S, Chen Z, Wang H et al (2015) Convolutional lstm network: a machine learning approach for precipitation nowcasting. In: Advances in neural information processing systems, pp 802–810
Xu Y, Gao L, Tian K et al (2019) Non-local convlstm for video compression artifact reduction. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 7043–7052
Xu Y, Zhao M, Liu J et al (2021) Boosting the performance of video compression artifact reduction with reference frame proposals and frequency domain information. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 213–222
Xue T, Chen B, Wu J et al (2019) Video enhancement with task-oriented flow. Int J Comput Vis 127(8):1106–1125
Yang R, Xu M, Wang Z et al (2018) Multi-frame quality enhancement for compressed video. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6664–6673
Yi P, Wang Z, Jiang K et al (2019) Progressive fusion video super-resolution network via exploiting non-local spatio-temporal correlations. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 3106–3115
Yue H, Cao C, Liao L et al (2020) Supervised raw video denoising with a benchmark dataset on dynamic scenes. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 2301–2310
Zamir SW, Arora A, Khan S et al (2020) Cycleisp: real image restoration via improved data synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 2696–2705
Zhang K, Luo W, Zhong Y et al (2018a) Adversarial spatio-temporal learning for video deblurring. IEEE Trans Image Process 28(1):291–301
Zhang R, Isola P, Efros AA et al (2018b) The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 586–595
Zhang K, Gool LV, Timofte R (2020) Deep unfolding network for image super-resolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 3217–3226

Zhao M, Xu Y, Zhou S (2021) Recursive fusion and deformable spatiotemporal attention for video compression artifact reduction. In: Proceedings of the 29th ACM international conference on multimedia, pp 5646–5654

Zhong Z, Gao Y, Zheng Y et al (2020) Efficient spatio-temporal recurrent neural network for video deblurring. In: European conference on computer vision. Springer, pp 191–207

Zhou S, Zhang J, Pan J et al (2019) Spatio-temporal filter adaptive network for video deblurring. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 2482–2491

Zhou K, Li W, Lu L et al (2022) Revisiting temporal alignment for video restoration. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 6053–6062

Zhu X, Hu H, Lin S et al (2019) Deformable convnets v2: More deformable, better results. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 9308–9316

Zhu C, Dong H, Pan J et al (2022) Deep recurrent neural network with multi-scale bi-directional propagation for video deblurring. In: Proceedings of the AAAI conference on artificial intelligence, pp 3598–3607