EDVR: Video Restoration with Enhanced Deformable Convolutional Networks

Xintao Wang\(^1\)  Kelvin C.K. Chan\(^2\)  Ke Yu\(^1\)  Chao Dong\(^3\)  Chen Change Loy\(^2\)

\(^1\)CUHK - SenseTime Joint Lab, The Chinese University of Hong Kong  \(^2\)Nanyang Technological University, Singapore  \(^3\)SIAT-SenseTime Joint Lab, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences

\{wx016, yk017\}@ie.cuhk.edu.hk  chao.dong@siat.ac.cn \{chan0899, ccloy\}@ntu.edu.sg

Abstract

Video restoration tasks, including super-resolution, deblurring, etc, are drawing increasing attention in the computer vision community. A challenging benchmark named REDS is released in the NTIRE19 Challenge. This new benchmark challenges existing methods from two aspects: (1) how to align multiple frames given large motions, and (2) how to effectively fuse different frames with diverse motion and blur. In this work, we propose a novel Video Restoration framework with Enhanced Deformable convolutions, termed EDVR, to address these challenges. First, to handle large motions, we devise a Pyramid, Cascading and Deformable (PCD) alignment module, in which frame alignment is done at the feature level using deformable convolutions in a coarse-to-fine manner. Second, we propose a Temporal and Spatial Attention (TSA) fusion module, in which attention is applied both temporally and spatially, so as to emphasize important features for subsequent restoration. Thanks to these modules, our EDVR wins the champions and outperforms the second place by a large margin in all four tracks in the NTIRE19 video restoration and enhancement challenges. EDVR also demonstrates superior performance to state-of-the-art published methods on video super-resolution and deblurring. The code is available at https://github.com/xinntao/EDVR.

1. Introduction

In this paper, we describe our winning solution in the NTIRE 2019 challenges on video restoration and enhancement. The challenge releases a valuable benchmark, known as RERealistic and Diverse Scenes dataset (REDS) [26], for the aforementioned tasks. In comparison to existing datasets, videos in REDS contain larger and more complex motions, making it more realistic and challenging. The competition enables fair comparisons among different algorithms and promotes the progress of video restoration.

Image restoration tasks such as super-resolution (SR) [5, 20, 41, 18, 45, 51] and deblurring [27, 15, 38] have experienced significant improvements over the last few years thanks to deep learning. The successes encourage the community to further attempt deep learning on the more challenging video restoration problems. Earlier studies [36, 4, 33, 19, 11] treat video restoration as a simple extension of image restoration. The temporal redundancy among neighboring frames is not fully exploited. Recent studies [2, 48, 37, 32] address the aforementioned problem with more elaborated pipelines that typically consist of four components, namely feature extraction, alignment, fusion, and reconstruction. The challenge lies in the design of the alignment and fusion modules when a video contains occlusion, large motion, and severe blurring. To obtain high-quality outputs, one has to (1) align and establish accurate correspondences among multiple frames, and (2) effectively fuse the aligned features for reconstruction.

Alignment. Most existing approaches perform alignment by explicitly estimating optical flow field between the reference and its neighboring frames [2, 48, 13]. The neighboring frames are warped based on the estimated motion fields. Another branch of studies achieve implicit motion compensation by dynamic filtering [10] or deformable convolution [40]. REDS imposes a great challenge to existing alignment algorithms. In particular, precise flow estimation and accurate warping can be challenging and time-consuming for flow-based methods. In the case of large motions, it is difficult to perform motion compensation either explicitly or implicitly within a single scale of resolution.

Fusion. Fusing features from aligned frames is another critical step in the video restoration task. Most existing methods either use convolutions to perform early fusion on all frames [2] or adopt recurrent networks to gradually fuse multiple frames [32, 6]. Liu et al. [22] propose a temporal adaptive network that can dynamically fuse across different
temporal scales. None of these existing methods consider the underlying visual informativeness on each frame — different frames and locations are not equally informative or beneficial to the reconstruction, as some frames or regions are affected by imperfect alignment and blurring.

Our Solution. We propose a unified framework, called EDVR, which is extensible to various video restoration tasks, including super-resolution and deblurring. The cores of EDVR are (1) an alignment module known as Pyramid, Cascading and Deformable convolutions (PCD), and (2) a fusion module known as Temporal and Spatial Attention (TSA).

The PCD module is inspired by TDAN [40] in using deformable convolutions to align each neighboring frame to the reference frame at the feature level. Different from TDAN, we perform alignment in a coarse-to-fine manner to handle large and complex motions. Specifically, we use a pyramid structure that first aligns features in lower scales with coarse estimations, and then propagates the offsets and aligned features to higher scales to facilitate precise motion compensation, similar to the notion adopted in optical flow estimation [7, 9]. Moreover, we cascade an additional deformable convolution after the pyramidal alignment operation to further improve the robustness of alignment.

The proposed TSA is a fusion module that helps aggregate information across multiple aligned features. To better consider the visual informativeness on each frame, we introduce temporal attention by computing the element-wise correlation between the features of the reference frame and each neighboring frame. The correlation coefficients then weigh each neighboring feature at each location, indicating how informative it is for reconstructing the reference image. The weighted features from all frames are then convolved and fused together. After the fusion with temporal attention, we further apply spatial attention to assign weights to each location in each channel to exploit cross-channel and spatial information more effectively.

We participated in all the four tracks in the video restoration and enhancement challenges [29, 28], including video super-resolution (clean/blur) and video deblurring (clean/compression artifacts). Thanks to the effective alignment and fusion modules, our EDVR has won the champion in all the four challenging tracks, demonstrating the effectiveness and the generalizability of our method. In addition to the competition results, we also report comparative results on existing benchmarks of video super-resolution and deblurring. Our EDVR shows superior performance to state-of-the-art methods in these video restoration tasks.

2. Related Work

Video Restoration. Since the pioneer work of SRCNN [5], deep learning methods have brought significant improvements in image and video super-resolution [20, 41, 18, 23, 49, 46, 2, 22, 32, 37, 48]. For video super-resolution, temporal alignment plays an important role and has been extensively studied. Several methods [2, 37, 32] use optical flow to estimate the motions between images and perform warping. However, accurate flow is difficult to obtain given occlusion and large motions. TOFlow [48] also reveals that the standard optical flow is not the optimal motion representation for video restoration. DUF [10] and TDAN [40] circumvent the problem by implicit motion compensation and surpass the flow-based methods. Our EDVR also enjoys the merits of implicit alignment, with a pyramid and cascading architecture to handle large motions.

Video deblurring also benefits from the development of learning-based methods [12, 24, 30, 34]. Several approaches [34, 50] directly fuse multiple frames without explicit temporal alignment, because the existence of blur increases the difficulty of motion estimation. Unlike these approaches, we attempt to acquire information from multiple frames using alignment, with a slight modification that an image deblurring module is added prior to alignment when there is a blur.

Deformable Convolution. Dai et al. [3] first propose deformable convolutions, in which additional offsets are learned to allow the network to obtain information away from its regular local neighborhood, improving the capability of regular convolutions. Deformable convolutions are widely used in various tasks such as video object detection [1], action recognition [52], semantic segmentation [3], and video super-resolution [40]. In particular, TDAN [40] uses deformable convolutions to align the input frames at the feature level without explicit motion estimation or image warping. Inspired by TDAN, our PCD module adopts deformable convolution as a basic operation for alignment.

Attention Mechanism. Attention has proven its effectiveness in many tasks [43, 47, 22, 23, 51]. For example, in video SR, Liu et al. [22] learn a set of weight maps to weigh the features from different temporal branches. Nonlocal operations [44] compute the response at a position as a weighted sum of the features at all positions for capturing long-range dependencies. Motivated by the success of these works, we employ both temporal and spatial attention in our TSA fusion module to allow different emphases on different temporal and spatial locations.

3. Methodology

3.1. Overview

Given $2N+1$ consecutive low-quality frames $I_{t-N:t+N}$, we denote the middle frame $I_t$ as the reference frame and the other frames as neighboring frames. The aim of video restoration is to estimate a high-quality reference frame $O_t$, which is close to the ground truth frame $O_t$. The overall framework of the proposed EDVR is shown in Fig. 2. It is a generic architecture suitable for several video restoration
tasks, including super-resolution, deblurring, denoising, de-blocking, etc.

Take video SR as an example, EDVR takes $2N+1$ low-resolution frames as inputs and generates a high-resolution output. Each neighboring frame is aligned to the reference one by the PCD align module at the feature level. The TSA fusion module fuses image information of different frames. The details of these two modules are described in Sec. 3.2 and Sec. 3.3. The fused features then pass through a reconstruction module, which is a cascade of residual blocks in EDVR and can be easily replaced by any other advanced modules in single image SR [46, 51]. The upsampling operation is performed at the end of the network to increase the spatial size. Finally, the high-resolution frame $O_t$ is obtained by adding the predicted image residual to a direct upsampled image.

For other tasks with high spatial resolution inputs, such as video deblurring, the input frames are first downsampled with strided convolution layers. Then most computation is done in the low-resolution space, which largely saves the computational cost. The upsampling layer at the end will resize the features back to the original input resolution. A PreDeblur module is used before the alignment module to pre-process blurry inputs and improve alignment accuracy.

Though a single EDVR model could achieve state-of-the-art performance, we adopt a two-stage strategy to further boost the performance in NTIRE19 competition. Specifically, we cascade the same EDVR network but with shallower depth to refine the output frames of the first stage. The cascaded network can further remove the severe motion blur that cannot be handled by the preceding model. The details are presented in Sec. 3.4.

### 3.2. Alignment with Pyramid, Cascading and Deformable Convolution

We first briefly review the use of deformable convolution for alignment [40], i.e., aligning features of each neighboring frame to that of the reference one. Differently from optical-flow based methods, deformable alignment is applied on features of each frame, denoted by $F_{t+i}, i \in [-N:+N]$. We use the modulated deformable module [53]. Given a deformable convolution kernel of $K$ sampling locations, we denote $w_k$ and $p_k$ as the weight and the pre-specified offsets for the $k$-th location, respectively. For instance, a $3 \times 3$ kernel is defined with $K=9$ and $p_k \in \{(-1,-1),(-1,0),\cdots,(1,1)\}$. The aligned features $F^m_{t+i}$ at each position $p_0$ can then be obtained by:

$$F^m_{t+i}(p_0) = \sum_{k=1}^{K} w_k \cdot F_{t+i}(p_0 + p_k + \Delta p_k) \cdot \Delta m_k.$$  \hspace{1cm} (1)

The learnable offset $\Delta p_k$ and the modulation scalar $\Delta m_k$ are predicted from concatenated features of a neighboring frame and the reference one:

$$\Delta P_{t+i} = f([F_{t+i}, F_i]), \quad i \in [-N:+N] \hspace{1cm} (2)$$

where $\Delta P = \{\Delta p\}$, $f$ is a general function consisting several convolution layers, and $[\cdot, \cdot]$ denotes the concatenation operation. For simplicity, we only consider learnable offsets $\Delta P_k$ and ignore modulation $\Delta m_k$ in the descriptions and figures. As $p_0 + p_k + \Delta p_k$ is fractional, bilinear interpolation is applied as in [3].

To address complex motions and large parallax problems in alignment, we propose PCD module based on well-established principles in optical flow: pyramidal processing [31, 35] and cascading refinement [7, 8, 9]. Specifically, as shown with black dash lines in Fig. 3, to generate feature $F^l_{t+i}$ at the $l$-th level, we use strided convolution filters to downsample the features at the $(l-1)$-th pyramid level by a factor of 2, obtaining $L$-level pyramids of feature representation. At the $l$-th level, offsets and aligned features are predicted also with the $\times 2$ upsampled offsets and aligned features from the upper $(l+1)$-th level, respectively (purple dash lines in Fig. 3):

$$\Delta P^{l}_{t+i} = f([F^{l}_{t+i}, F^{l}_{i}], (\Delta P^{l+1}_{t+i})^{\times 2}),$$  \hspace{0.5cm} (3)

$$(F^{m l}_{t+i})^{l} = g(DConv((F^{l}_{t+i}, \Delta P^{l}_{t+i}), ((F^{m l+1}_{t+i})^{\times 1})^{\times 2}),$$  \hspace{0.5cm} (4)
where \((\cdot)^\uparrow s\) refers to upscaling by a factor \(s\), \(\text{DConv}\) is the deformable convolution described in Eqn. 1, and \(g\) is a general function with several convolution layers. Bilinear interpolation is adopted to implement the \(\times 2\) upscaling. We use three-level pyramid, \(i.e., L=3\), in EDVR. To reduce computational cost, we do not increase channel numbers as spatial sizes decrease.

Following the pyramid structure, a subsequent deformable alignment is cascaded to further refine the coarsely aligned features (the part with light purple background in Fig. 3). PCD module in such a coarse-to-fine manner improves the alignment to the sub-pixel accuracy. We demonstrate the effectiveness of PCD in Sec. 4.3. It is noteworthy that the PCD alignment module is jointly learned together with the whole framework, without additional supervision [40] or pretraining on other tasks like optical flow [48].

### 3.3. Fusion with Temporal and Spatial Attention

Inter-frame temporal relation and intra-frame spatial relation are critical in fusion because 1) different neighboring frames are not equally informative due to occlusion, blurry regions and parallax problems; 2) misalignment and unalignment arising from the preceding alignment stage adversely affect the subsequent reconstruction performance. Therefore, dynamically aggregating neighboring frames in pixel-level is indispensable for effective and efficient fusion. In order to address the above problems, we propose TSA fusion module to assign pixel-level aggregation weights on each frame. Specifically, we adopt temporal and spatial attentions during the fusion process, as shown in Fig. 4.

The goal of temporal attention is to compute frame similarity in an embedding space. Intuitively, in an embedding space, a neighboring frame that is more similar to the reference one, should be paid more attention. For each frame \(i\in[-N;+N]\), the similarity distance \(h\) can be calculated as:

\[
h(F_{t+i}^a, F_t^a) = \text{sigmoid}(\theta(F_{t+i}^a)^T\phi(F_t^a)),
\]

where \(\theta(F_{t+i}^a)\) and \(\phi(F_t^a)\) are two embeddings, which can be achieved with simple convolution filters. The sigmoid activation function is used to restrict the outputs in \([0, 1]\), stabilizing gradient back-propagation. Note that for each spatial location, the temporal attention is spatial-specific, \(i.e.,\) the spatial size of \(h(F_{t+i}^a, F_t^a)\) is the same as that of \(F_{t+i}^a\).

The temporal attention maps are then multiplied in a pixel-wise manner to the original aligned features \(F_{t+i}^a\). An extra fusion convolution layer is adopted to aggregate these attention-modulated features \(\tilde{F}_{t+i}^F\):

\[
\tilde{F}_{t+i}^F = \tilde{F}_{t+i}^a \odot h(F_{t+i}^a, F_t^a),
\]

where \(\odot\) and \([\cdot, \cdot, \cdot]\) denote the element-wise multiplication and concatenation, respectively.

Spatial attention masks are then computed from the fused features. A pyramid design is employed to increase the attention receptive field. After that, the fused features are modulated by the masks through element-wise multiplication and addition, similar to [45]. The effectiveness of TSA module is presented in Sec. 4.3.

### 3.4. Two-Stage Restoration

Though a single EDVR equipped with PCD alignment module and TSA fusion module could achieve state-of-the-art performance, it is observed that the restored images are
not perfect, especially when the input frames are blurry or severely distorted. In such a harsh circumstance, motion compensation and detail aggregation are affected, resulting in inferior reconstruction performance.

Intuitively, coarsely restored frames would greatly mitigate the pressure for alignment and fusion. Thus, we employ a two-stage strategy to further boost the performance. Specifically, a similar but shallower EDVR network is cascaded to refine the output frames of the first stage. The benefits are two-fold: 1) it effectively removes the severe motion blur that cannot be handled in the preceding model, improving the restoration quality; 2) it alleviates the inconsistency among output frames. The effectiveness of two-stage restoration is illustrated in Sec. 4.4.

4. Experiments

4.1. Training Datasets and Details

Training datasets. Previous studies on video processing [21, 10, 34] are usually developed or evaluated on private datasets. The lack of standard and open video datasets restricts fair comparisons. REDS [26] is a newly proposed high-quality (720p) video dataset in the NTIRE19 Competition. REDS consists of 240 training clips, 30 validation clips and 30 testing clips (each with 100 consecutive frames). During the competition, since the test ground truth is not available, we select four representative clips (with diverse scenes and motions) as our test set, denoted by REDS4. The remaining training and validation clips are re-grouped as our training dataset (a total of 266 clips). To be consistent with our methods and process in the competition, we also adopt this configuration in this paper.

Vimeo-90K [48] is a widely used dataset for training, usually along with Vid4 [21] and Vimeo-90K testing dataset (denoted by Vimeo-90K-T) for evaluation. We observe dataset bias when the distribution of training sets deviates from that of testing sets. More details are presented in Sec. 4.3.

Training details. The PCD alignment module adopts five residual blocks (RB) to perform feature extraction. We use 40 RBs in the reconstruction module and 20 RBs for the second-stage model. The channel size in each residual block is set to 128. We use RGB patches of size 64×64 and 256×256 as inputs for video SR and deblurring tasks, respectively. Mini-batch size is set to 32. The network takes five consecutive frames (i.e., N=2) as inputs unless otherwise specified. We augment the training data with random horizontal flips and 90° rotations. We only adopt Charbonnier penalty function [17] as the final loss, defined by

\[ L = \sqrt{\| \hat{O}_t - O_t \|^2 + \varepsilon^2} \], where \( \varepsilon \) is set to \( 1 \times 10^{-3} \).

We train our model with Adam optimizer [14] by setting \( \beta_1 = 0.9 \) and \( \beta_2 = 0.999 \). The learning rate is initialized as \( 4 \times 10^{-4} \), We initialize deeper networks by parameters from shallower ones for faster convergence. We implement our models with the PyTorch framework and train them using 8 NVIDIA Titan Xp GPUs.

4.2. Comparisons with State-of-the-art Methods

We compare our EDVR with several state-of-the-art methods on video SR and video deblurring respectively. Two-stage and self-ensemble strategies [20] are not used. In the evaluation, we include all the input frames and do not crop any border pixels except the DUF method [10]. We crop eight pixels near image boundary for DUF due to its severe boundary effects.

Video Super-Resolution. We compare our EDVR method with nine algorithms: RCAN [51], DeepSR [19], BayesSR [21], VESPCN [2], SPMC [37], DeblurGAN [48], FRVSR [32], DUF [10] and RBPN [6] on three testing datasets: Vid4 [21], Vimeo-90K-T [48] and REDS4. Most previous methods use different training sets and different down-sampling kernels, making the comparisons difficult. Each testing dataset has different characteristics. Vid4 is commonly used in video SR. The data has limited motion. Visual artifacts also exist on its ground-truth (GT) frames. Vimeo-90K-T is a much larger dataset with various motions and diverse scenes. REDS4 consists of high-quality images but with larger and more complex motions. We observe dataset bias when training and testing sets diverge a lot. Hence, we train our models on Vimeo-90K when evaluated on Vid4 and Vimeo-90K-T.

The quantitative results on Vid4, Vimeo-90K-T and REDS4 are shown in Table 1, Table 2 and Table 3 (Left), respectively. On Vid4, EDVR achieves comparable performance to DUF and outperforms other methods by a large margin. On Vimeo-90K-T and REDS, EDVR is significantly better than the state-of-the-art methods, including DUF and RBPN. Qualitative results on Vid4 and Vimeo-90K-T are presented in Fig. 5 and Fig. 6, respectively. On both datasets, EDVR recovers more accurate textures compared to existing methods, especially in the second image of Fig. 6, where the characters can be correctly identified only in the outputs of EDVR.

Video Deblurring. We compare our EDVR method with four algorithms: DeepDeblur[27], DeblurGAN [16], SRN-Deblur [39] and DBN [34] on the REDS4 dataset. Quantitative results are shown in Table 3 (Right). Our EDVR outperforms the state-of-the-art methods by a large margin. We attribute this to both the effectiveness of our method and the challenging REDS dataset that contains complex blurring. Visual results are presented in Fig. 7, while most methods are able to address small blurring, only EDVR can successfully recover clear details from extremely blurry images.
Table 1. Quantitative comparison on Vid4 for 4x video SR. Red and blue indicates the best and the second best performance, respectively. Y or RGB denotes the evaluation on Y (luminance) or RGB channels. * indicates the values are taken from their publications.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Calendar (Y)</td>
<td>20.39/0.7570</td>
<td>22.33/0.7254</td>
<td>-</td>
<td>-</td>
<td>22.16/0.7465</td>
<td>22.47/0.7318</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>City (Y)</td>
<td>25.16/0.6028</td>
<td>26.10/0.6960</td>
<td>27.00/0.7573</td>
<td>26.78/0.7403</td>
<td></td>
<td>28.27/0.8313</td>
<td>27.73/0.8035</td>
<td>28.00/0.8122</td>
<td></td>
</tr>
<tr>
<td>Foliage (Y)</td>
<td>23.47/0.5666</td>
<td>24.74/0.6647</td>
<td>-</td>
<td>25.43/0.7208</td>
<td>25.27/0.7092</td>
<td></td>
<td>26.41/0.7709</td>
<td>26.22/0.757</td>
<td>26.34/0.7635</td>
</tr>
<tr>
<td>Walk (Y)</td>
<td>26.10/0.7974</td>
<td>28.65/0.8719</td>
<td>-</td>
<td>28.91/0.8761</td>
<td>29.05/0.8790</td>
<td>30.60/0.9141</td>
<td>30.70/0.909</td>
<td>31.02/0.9152</td>
<td></td>
</tr>
<tr>
<td>Average (Y)</td>
<td>23.78/0.6347</td>
<td>25.46/0.7395</td>
<td>25.35/0.7557</td>
<td>25.88/0.7752</td>
<td>26.39/0.8282</td>
<td>27.33/0.8316</td>
<td>27.32/0.8245</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average (RGB)</td>
<td>22.37/0.6098</td>
<td>24.02/0.7192</td>
<td>-</td>
<td>24.39/0.7334</td>
<td>24.41/0.7428</td>
<td>-</td>
<td>25.79/0.8136</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

Table 2. Quantitative comparison on Vimeo-90K-T for 4x video SR. * indicates the values are taken from [48]. ** indicates the values are taken from their publications.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RGB Channels</td>
<td>29.79/0.8483</td>
<td>33.61/0.9101</td>
<td>33.55/0.8948</td>
<td>24.64/0.8205</td>
<td>33.08/0.9054</td>
<td>34.33/0.9227</td>
<td>36.37/0.9387</td>
<td>37.07/0.9435</td>
</tr>
<tr>
<td>Y Channel</td>
<td>31.32/0.8684</td>
<td>35.35/0.9251</td>
<td>25.55/0.8498</td>
<td>24.64/0.8205</td>
<td>33.08/0.9054</td>
<td>34.33/0.9227</td>
<td>36.37/0.9387</td>
<td>37.07/0.9435</td>
</tr>
</tbody>
</table>

Table 3. Quantitative comparison on REDS4. Left: 4x Video SR (clean); Right: Video deblurring (clean). Test on RGB channels.

| Method | Clip000 | Clip011 | Clip015 | Clip020 | Average | Clip0200 | Clip0211 | Clip0215 | Clip0210 | Average | Clip0300 | Clip0311 | Clip0315 | Clip0320 | Average |
|--------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| Bicubic | 24.55/0.6489 | 26.06/0.7265 | 28.52/0.8034 | 25.41/0.7360 | 26.14/0.7292 | 26.76/0.8258 | 26.47/0.8238 | 26.36/0.8252 | 26.49/0.8248 | 26.09/0.8248 | 26.36/0.8252 | 26.47/0.8238 | 26.36/0.8252 | 26.49/0.8248 |
| RCAN [51] | 26.17/0.7371 | 29.34/0.8235 | 31.95/0.8811 | 27.74/0.8293 | 28.75/0.8200 | 29.13/0.9028 | 24.26/0.7656 | 28.50/0.8229 | 26.66/0.6492 | 27.16/0.8248 |
| TOFlow [48] | 28.52/0.7507 | 28.80/0.7858 | 30.67/0.8609 | 26.92/0.7953 | 27.98/0.7990 | 28.95/0.8734 | 25.46/0.7959 | 29.26/0.8706 | 32.61/0.9018 |
| DUF [10] | 27.90/0.7937 | 28.38/0.8056 | 31.55/0.8846 | 27.30/0.8164 | 28.63/0.8251 | 32.00/0.9018 | 24.28/0.7313 | 29.00/0.8878 | 25.10/0.7035 | 26.35/0.8066 |
| EDVR (Ours) | 28.01/0.8252 | 30.17/0.8864 | 34.06/0.9206 | 30.09/0.8881 | 31.09/0.8800 | 34.33/0.9393 | 36.09/0.9452 | 34.80/0.9487 |

Figure 5. Qualitative comparison on the Vid4 dataset for 4x video SR. Zoom in for best view.

Figure 6. Qualitative comparison on the Vimeo-90K-T dataset for 4x video SR. Zoom in for best view.

Figure 7. Qualitative comparison on the REDS4 dataset for video deblurring. Zoom in for best view.
Table 4. Ablations on: **Left**: PCD and TSA modules (Experiments here adopt a smaller model with 10 RBs in the reconstruction module and the channel number is set to 64). FLOPs [25] are calculated on an image with the HR size of $1280 \times 720$. **Right**: the bias between the training and testing datasets.

<table>
<thead>
<tr>
<th>Model</th>
<th>Model 1</th>
<th>Model 2</th>
<th>Model 3</th>
<th>Model 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCD?</td>
<td>✓ (1 DConv)</td>
<td>✓ (4 DConv)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>TSA?</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>PSNR</td>
<td>29.78</td>
<td>29.98</td>
<td>30.39</td>
<td>30.53</td>
</tr>
<tr>
<td>FLOPs</td>
<td>640.2G</td>
<td>932.9G</td>
<td>939.3G</td>
<td>936.5G</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Test</th>
<th>REDS4</th>
<th>Vid4 [21]</th>
<th>Vimeo90k [48]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REDS (5 frames)</td>
<td>31.09/0.8800</td>
<td>25.37/0.7956</td>
<td>34.33/0.9246</td>
</tr>
<tr>
<td>Vimeo-90K (7 frames)</td>
<td>30.49/0.8700</td>
<td>25.83/0.8077</td>
<td>35.79/0.9374</td>
</tr>
<tr>
<td>$\Delta$</td>
<td>0.60/0.0100</td>
<td>-0.46/-0.0121</td>
<td>-1.46/-0.0128</td>
</tr>
</tbody>
</table>

Figure 8. Ablation on the PCD alignment module. Compared with the results without PCD alignment, the flow of the PCD outputs is much smaller and cleaner, indicating that the PCD module can successfully handle large and complex motions. **Flow field color coding scheme** is shown in the right. The direction and magnitude of the displacement vector are represented by hue and color intensity, respectively.

Figure 9. Ablation on the TSA fusion module. The frames and regions of lower flow magnitude tend to have more attention, indicating that the corresponding frames and regions are more informative.

### 4.3. Ablation Studies

**PCD Alignment Module.** As shown in Table 4 (Left), our baseline (Model 1) only adopts one deformable convolution for alignment. Model 2 follows the design of TDAN [40] to use four deformable convolutions for alignment, achieving an improvement of 0.2 dB. With our proposed PCD module, Model 3 is nearly 0.4 dB better than Model 2 with roughly the same computational cost, demonstrating the effectiveness of PCD alignment module. In Fig. 8, we show representative features before and after different alignment modules, and depict the flow (derived by PWCNet [35]) between reference and neighboring features. Compared with the flow without PCD alignment, the flow of the PCD outputs is much smaller and cleaner, indicating that the PCD module can successfully handle large and complex motions.

**TSA Attention Module.** As shown in Table 4 (Left), with the TSA attention module, Model 4 achieves 0.14 dB performance gain compared to Model 3 with similar computations. In Fig. 9, we present the flow between the reference and neighboring frames, together with the temporal atten-
Figure 10. Qualitative results of our EDVR method on the four tracks in the NTIRE 2019 video restoration and enhancement challenges.

Table 5. Top5 methods in the NTIRE 2019 challenges on video restoration and enhancement. Red and blue indicates the best and the second best performance, respectively.

<table>
<thead>
<tr>
<th>Method</th>
<th>SR</th>
<th>Blur</th>
<th>Deblur</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDVR (Ours)</td>
<td>31.79/0.8962</td>
<td>30.17/0.8647</td>
<td>36.96/0.9657</td>
</tr>
<tr>
<td>2nd method</td>
<td>31.13/0.8811</td>
<td>-/-</td>
<td>35.71/0.9522</td>
</tr>
<tr>
<td>3rd method</td>
<td>31.00/0.8822</td>
<td>27.71/0.8067</td>
<td>34.99/0.9361</td>
</tr>
<tr>
<td>4th method</td>
<td>30.97/0.8804</td>
<td>28.92/0.8333</td>
<td>33.71/0.9363</td>
</tr>
<tr>
<td>5th method</td>
<td>30.91/0.8782</td>
<td>28.96/0.8307</td>
<td>33.46/0.9293</td>
</tr>
</tbody>
</table>

4.4. Evaluation on REDS Dataset

We participated in all the four tracks in the NTIRE19 video restoration and enhancement challenges [29, 28]. Quantitative results are presented in Table 5. Our EDVR wins the champions and outperforms the second place by a large margin in all tracks. In the competition, we adopt self-ensemble as [42, 20]. Specifically, during the test time, we flip and rotate the input image to generate four augmented inputs for each sample. We then apply the EDVR method on each, reverse the transformation on the restored outputs and average for the final result. The two-stage restoration strategy as described in Sec. 3.4 is also used to boost the performance. As shown in Table 6, we observe that the two-stage restoration largely improves the performance around 0.5 dB (EDVR(+) vs. EDVR-S2(+)). While the self-ensemble is helpful in the first stage (EDVR vs. EDVR+), it only brings marginal improvement in the second stage (EDVR-S2 vs. EDVR-S2+). Qualitative results are shown in Fig. 10. It is observed that the second stage helps recover clear details in challenging cases, e.g., the inputs are extremely blurry.

5. Conclusion

We have introduced our winning approach in the NTIRE 2019 video restoration and enhancement challenges. To handle the challenging benchmark released in the competition, we propose EDVR, a unified framework with unique designs to achieve good alignment and fusion quality in various video restoration tasks. Thanks to the PCD alignment module and TSA fusion module, EDVR not only wins all four tracks in the NTIRE19 Challenges but also demonstrates superior performance to existing methods on several benchmarks of video super-resolution and deblurring.

Acknowledgement. We thank Yapeng Tian for providing the core codes of TDAN [40]. This work is supported by SenseTime Group Limited, Joint Lab of CAS-HK, the General Research Fund sponsored by the Research Grants Council of the Hong Kong SAR (CUHK 14241716, 1422416, 14209217), and Singapore MOE AcRF Tier 1 (M4012082.020).
References


