A Codec Information Assisted Framework for Efficient Compressed Video Super-Resolution

Hengsheng Zhang[1], Xueyi Zou[2], Jiaming Guo[2], Youliang Yan[2], Rong Xie[1], and Li Song[1,3]

[1] Institute of Image Communication and Network Engineering, Shanghai Jiao Tong University
[2] Huawei Noah's Ark Lab
[3] MoE Key Lab of Artifical Intelligence, AI Institute, Shanghai Jiao Tong University
{hs_zhang,xierong,song_li}@sjtu.edu.cn
{zouxueyi,guojiaming5,yanyouliang}@huawei.com

Abstract. Online processing of compressed videos to increase their resolutions attracts increasing and broad attention. Video Super-Resolution (VSR) using recurrent neural network architecture is a promising solution due to its efficient modeling of long-range temporal dependencies. However, state-of-the-art recurrent VSR models still require significant computation to obtain a good performance, mainly because of the complicated motion estimation for frame/feature alignment and the redundant processing of consecutive video frames. In this paper, considering the characteristics of compressed videos, we propose a Codec Information Assisted Framework (CIAF) to boost and accelerate recurrent VSR models for compressed videos. Firstly, the framework reuses the coded video information of Motion Vectors to model the temporal relationships between adjacent frames. Experiments demonstrate that the models with Motion Vector based alignment can significantly boost the performance with negligible additional computation, even comparable to those using more complex optical flow based alignment. Secondly, by further making use of the coded video information of Residuals, the framework can be informed to skip the computation on redundant pixels. Experiments demonstrate that the proposed framework can save up to 70% of the computation without performance drop on the REDS4 test videos encoded by H.264 when CRF is 23.

Keywords: Efficient video super-resolution, Compressed video, Codec information assisted, Motion Vectors, Residuals

1 Introduction

Compressed videos are prevalent on the Internet, ranging from movies, webcasts to user-generated videos, most of which are of relatively low resolutions and qualities. Many terminal devices, such as smartphones, tablets, and TVs, come with a 2K/4K or even 8K definition screen. Thus, there is an urgent demand for

such devices to be able to online super-resolve the low-resolution videos to the resolution of the screen definition. Video Super-Resolution (VSR) increases the video frames' resolution by exploiting redundant and complementary information along the video temporal dimension. With the wide use of neural networks in computer vision tasks, on the one hand, neural network based VSR methods outperform traditional ones. But on the other hand, they require a lot of computation and memory, which current commercial terminal devices cannot easily provide.

Most neural network based VSR models come with a lot of repeated computation or memory consumption. For example, sliding-window based VSR models[10,25,27,5] have to extract the features of adjacent frames repeatedly. Although this process can be optimized by preserving the feature maps of previous frames, it increases memory consumption. Besides, to make the most of adjacent frames' information, frame alignment is an essential part of many such models, which is usually implemented by optical flow prediction[21,24], deformable convolution[6,34], attention/correlation[16], and other complicated modules[13,32]. This frame alignment process also increases model complexity, and many of the operators are not well supported by current terminal chipsets.

Many VSR methods use recurrent neural networks to avoid repeated feature extraction and to exploit long-range dependencies. The previous frame's high-resolution information (image or features) is reused for the current frame prediction. Several information propagation schemes have been proposed, such as unidirectional propagation[23,8,11], bidirectional propagation[2,17], and the more complex grid propagation[3,31]. As expected, the more complex the propagation scheme is, the better the super-resolution performs in terms of PSNR/SSIM or visual quality. However, considering the stringent computational budget of terminal devices and the online processing requirement, most complex propagation schemes, such as bidirectional propagation and grid propagation, are not good choices. Unidirectional recurrent models seem to be good candidates, but to get better performance, frame/feature alignment is also indispensable. As mentioned above, mainstream methods for alignment are computationally heavy and not well supported by current terminal chipsets.

Compared with raw videos, compressed videos have some different characteristics. When encoding, the motion relationships of the current frame and a reference frame (e.g. the previous frame) are calculated as **Motion Vectors** (MVs). The reference frame is then warped according to MVs to get the predicted image of the current time step. The differences between the predicted image and current frame are calculated as **Residuals**. MVs and Residuals are encoded in the video streams, with MVs providing motion cues of video frames and Residuals indicating the motion-compensated differences between frames. When decoding, MVs and Residuals are extracted to rebuild the video frames sequentially based on the previous rebuilt frames.

By leveraging the characteristics of compressed videos, we propose a Codec Information Assisted Framework (CIAF) to improve the performance and the efficiency of unidirectional recurrent VSR methods. To align the features of pre-

vious frame, we reuse the MVs to model the temporal relationships between adjacent frames. The models using MV-based alignment can significantly boost the performance with negligible additional computation, even reaching a comparable performance with those using more complex optical flow based alignment. To further reduce terminal device computation burden, we apply most computation (convolutions) only to changed regions of consecutive frames. For the rest areas, we reuse features of the previous frame by warping part of the feature maps generated in the last step according to MVs. The way to determine where the change happens is based on Residuals, i.e., only pixels with Residuals not equal to zero are considered to be changed. Due to the high degree of similarity between video frames, the proposed approach can skip lots of computation. The experiments show up to 70% of computation can be saved without performance drop on the REDS4 [27] test videos encoded by H.264 when CRF is 23.

The contributions of this paper can be summarized as follows. (1) We propose to reuse the coded video information of MVs to model temporal relationships between adjacent frames for frame/feature alignment. Models with MV-based alignment can significantly boost performance with minimal additional computation, even matching the performance of optical flow based models. (2) We find that the coded information of Residuals can inform the VSR models to skip the computation on redundant pixels. The models using Residual-informed sparse processing can save lots of computation without a performance drop. (3) We disclose some of the crucial tricks to train the CIAF, and we evaluate some of the essential design considerations contributing to the efficient compressed VSR model.

2   Related Work

In this section, we first review the CNN-based video super-resolution work. Then, we discuss adaptive CNN acceleration techniques related to our work.

2.1   Video Super-Resolution

Video super-resolution (VSR) is challenging because complementary information must be aggregated across misaligned video frames for restoration. There are mainly two forms of VSR algorithms: sliding-window methods and recurrent methods.

Sliding-window methods. Sliding-window methods restore the target high-resolution frame from the current and its neighboring frames. [1,30] align the neighboring frames to the target frame with predicted optical flows between input frames. Instead of explicitly aligning frames, RBPN[10] treats each context frame as a separate source of information and employs back-projection for iterative refining of target HR features. DUF[13] utilizes generated dynamic upsampling filters to handle motions implicitly. Besides, deformable convolutions (DCNs)[6,34] are introduced to express temporal relationships. TDAN[25] aligns neighboring frames with DCNs in the feature space. EDVR[27] uses DCNs

on a multi-scale basis for more precise alignment. MuCAN[16] searches similar patches around the target position from neighboring frames instead of direct motion estimation. [5] extracts Motion Vectors from compressed video streams as motion priors for alignment and incorporates coding priors into modified SFT blocks[28] to refine the features from the input LR frames. These methods can produce pleasing results, but they are challenging to be applied in practice on the terminal devices due to repeated feature extraction or complicated motion estimation.

Recurrent methods. Unlike sliding-window methods, recurrent methods take the output of the past frame processing as a prior input for the current iteration. So the recurrent networks are not only efficient but also can take account of long-range dependencies. In unidirectional recurrent methods FRVSR[23], RLSP[8] and RSDN[11], information is sequentially propagated from the first frame to the last frame, so this kind of scheme has the potential to be applied for online processing. Besides, FRVSR[23] aligns the past predicted HR frame with optical flows for the current iteration. RLSP[8] and RSDN[11] employs high-dimensional latent states to implicitly transfer temporal information between frames. Different from unidirectional recurrent networks, BasicVSR[2] proposes a bidirectional propagation scheme to better exploit temporal features. BasicVSR++[3] redesigns BasicVSR by proposing second-order grid propagation and flow-guided deformable alignment. Similar with BasicVSR++, [31] employs complex grid propagation to boost the performance. COMISR[17] applies a bidirectional recurrent model to compressed video super-resolution and uses a CNN to predict optical flows for alignment. Although they can achieve state-of-the-art performance, the complicated information propagation scheme and complex motion estimation make them unpractical to apply to the terminal device with online processing.

2.2   Adaptive Inference

Most of the existing CNN methods treat all regions in the image equally. But the flat area is naturally easier to process than regions with textures. Adaptive inference can adapt the network structure according to the characteristics of the input. BlockDrop[29] proposes to dynamically pick which deep network layers to run during inference to decrease overall computation without compromising prediction accuracy. ClassSR[14] uses a "class module" to decompose the image into sub-images with different reconstruction difficulties and then applies networks with various complexity to process them separately. Liu et al. [19] establishes adaptive inference for SR by adjusting the number of convolutional layers used at various locations. Wang et al. [26] locate redundant computation by predicted spatial and channel masks and use sparse convolution to skip redundant computation. The image-based acceleration algorithms follow the internal characteristics of images, so they can only reduce spatial redundancy.

Most of the time, the changes between consecutive frames in a video are insignificant. Based on this observation, Skip-Convolutions[9] limits the computation only to the regions with significant changes between frames while skipping
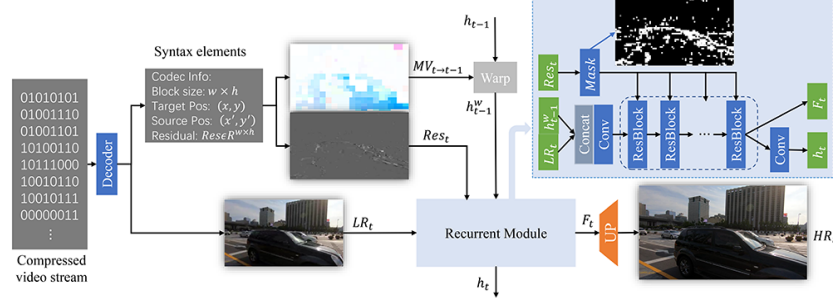
Fig. 1: Overview of the proposed codec information assisted framework (CIAF). The h_{t-1} is the refined features from past frame LR_{t-1}. Motion Vector ($MV_{t \to t-1}$) and Residuals (Res_t) are the codec information. In our model, we utilize the Motion Vector to align the features from the past frame. Besides, the sparse processing is applied in the Resblocks only to calculate the regions with Residuals.

the others. But this model is primarily applicable to high-level tasks. FAST[33], the most similar work with ours, employs SRCNN[7] to only generate the HR image of the first frame in a group of frames. In the following iterations, the HR blocks of the last frame are transferred to the current frame according to MVs. Finally, the up-sampled Residuals are added to the transferred HR image to generate the HR output of the current frame. The operations are on the pixel level, which can easily lead to errors. Instead of directly reusing the HR pixels from past frames, we utilize MVs to conduct an efficient alignment for unidirectional recurrent VSR systems. And the Residuals are used to determine the locations of redundancy.

3   Codec Information Assisted Framework

In this section, we first introduce the basics of video coding related to our framework. Then we present our codec information assisted framework (CIAF, Fig. 1) consisting of two major parts, i.e., the Motion Vector (MV) based alignment and Residual informed sparse processing.

3.1   Video coding Basics

The Inter-Prediction Mode (Fig. 2) of video codec inspires our framework. Generally, there is a motion relationship between the objects in each frame and its adjacent frames. The motion relationship of this kind of object constitutes the temporal redundancy between frames. In H.264[22], temporal redundancy is reduced by motion estimation and motion compensation. As Fig. 2 shows, in motion estimation, for every current block, we can find a similar pixel block as a reference in the reference frame. The relative position between the current

pixel block in the current frame and the reference block in the reference frame is represented by (MV_x, MV_y), a vector of two coordinate values used to indicate this relative position, known as the **Motion Vector (MV)**. In motion compensation, we use the found reference block as a prediction of the current block. Because there are slight differences between the current and reference blocks, the encoder needs to calculate the differences as **Residual**. When decoding, we first use the decoded reference frame and MVs to generate the prediction image of the target frame. Then we add decoded Residuals to the prediction image to get the target frame. In our paper, we reuse the MVs and Residuals to increase the efficiency of unidirectional recurrent VSR models.
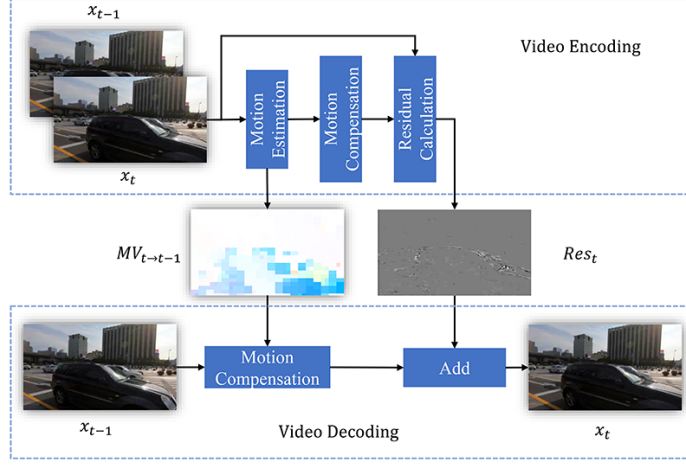

Fig. 2: The Inter-Prediction Mode of video codec.

3.2   Motion Vector based Alignment

In VSR methods, alignment between neighboring frames is important for good performance. In this paper, for alignment, we warp the HR information of the past frame with MVs. Different from the interpolation filter used in H.264, the bilinear interpolation filter is applied to the pixels for efficiency if the MV is fractional. When there is an insufficient temporal connection between blocks, the video encoder utilizes intra-prediction. Since the intra-blocks mainly appear in the keyframe (the first frame of a video clip) and there are few intra-predicted blocks in most frames, for blocks with intra-prediction, we transfer the features of the same position in the adjacent frame. To a common format, we set $MV = (0,0)$ for intra-blocks. We can formulate a motion field MV with size $H \times W \times 2$ like optical flow. H and W are the height and width of the input LR frame, respectively. The third dimension indicates the relative position

in the width and height directions. So the MV is an approximate alternative to optical flow. In this way, we bypass the complicated motion estimation. The MV-based alignment can boost the performance of existing unidirectional recurrent VSR models and even achieve comparable performance with optical flow based alignment, as demonstrated later.

3.3   Residual Informed Sparse Processing

As Fig. 1 shows, in the paper, we design a Residual informed sparse processing to reduce redundant computation. Residuals represent the difference between the warped frame and the current frame. The areas without Residuals indicate the current region can be directly predicted by sharing the corresponding patches from the reference frame. Therefore, Residuals can locate the areas that need to be further refined. With the guide of Residuals, we only make convolutions on the "important" pixels. The features of the rest pixels are enhanced by aggregation with the MV-warped features from the past frame. As Fig. 1 shows, to make it robust, we adopt this sparse processing to the body (Resblocks) of the network, the head and tail Conv layers are applied on all pixels.

Benifict from motion estimation and motion compensation, we can easily predict the flat regions or regular structures like brick wall for current frame according to the contents of adjacent frames without loss (Residuals). Residuals are more likely to be introduced on complex textures. Because flat regions or regular structures take up the majority of the frame, Residuals are sparse in most scenes. Based on these characteristics, the proposed Residual informed sparse processing can significantly reduce the space-time redundancy computation while maintaining the comparable performance with baseline.
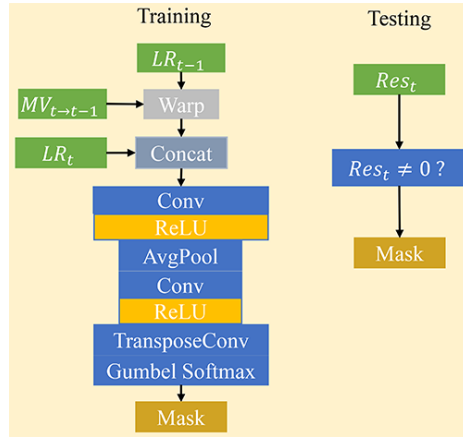

Fig. 3: The sparse mask generation. Res_t is the Residual extracted from compressed video. When training, we use a tiny CNN to predict a spatial mask; when testing, convolutions are only applied to pixels whose Residual is not equal to 0.

Because the Residuals are sparse, only a tiny part of pixels optimize the model if we directly utilize Residuals to decide where to conduct convolutions during training. In experiments, we find it hard to converge. We design a Simulated Annealing strategy to slowly reduce the number of pixels involved in training, which is a critical trick in our sparse processing. As Fig. 3 shows, we utilize a light CNN model to identify the changed regions according to the current frame and the MV-warped past frame. Following [26], Gumbel softmax trick[12] is used to produce a spatial mask $M \in R^{H \times W}$ with the output features $F \in R^{2 \times H \times W}$.

$$M[x,y] = \frac{exp((F[1,x,y] + G[1,x,y])/\tau)}{\sum_{i=1}^{2} exp((F[i,x,y] + G[i,x,y])/\tau)} \tag{1}$$

where x and y are vertical and horizontal indices, $G \in R^{2 \times H \times W}$ is a Gumbel noise vector with all elements following $Gumbel(0,1)$ distribution and τ is the temperature parameter. Samples from Gumbel softmax distribution become uniform if $\tau \to \infty$. When $\tau \leftarrow 0$, samples from Gumbel softmax distribution become one-hot. The predicted mask gradually becomes sparse with training.
Training Strategy: During training, we utilize a sparsity regularization loss to supervise the model:

$$L_{reg} = \frac{1}{H \times W} \sum_{h,w} M[w,h] \tag{2}$$

According the Simulated Annealing strategy, we set the weight of L_{reg}:

$$\lambda = min(\frac{t}{T_{epoch}}, 1) \cdot \lambda_0 \tag{3}$$

where t is the current number of epochs, T_{epoch} is empirically set to 20, and λ_0 is set to 0.004. And the temperature parameter τ in the Gumbel softmax trick is initialized as 1 and gradually decreased to 0.5:

$$\tau = max(1 - \frac{t}{T_{temp}}, 0.5) \tag{4}$$

where T_{temp} is set to 40 in this paper.
Testing: When testing, we directly replace the mask-prediction CNN with Residuals to select the pixels to calculate. This process is formulated as:

$$M_{test}[x,y] = (Res[x,y] \neq 0) \tag{5}$$

where $Res[x,y]$ represents the Residual value at position $[x,y]$. When Residual is equal to 0, the pixel is skipped.

4   Experiments

4.1   Implementation Details

We use dataset REDS[20] for training. REDS dataset has large motion between consecutive frames captured from a hand-held device. We evaluate the networks

on the datasets REDS4[27] and Vid4[18]. All frames are first smoothed by a Gaussian kernel with the width of 1.5 and downsampled by 4. Because our framework is designed for compressed videos, we further encode the datasets with H.264[22], the most common video codec, at different compression rates. The recommended CRF value in H.264 is between 18 and 28, and the default is 23. In experiments, we set CRF values to 18, 23, and 28 and use the FFmpeg codec to encode the datasets.

Our goal is to design efficient and online processing VSR systems, so we do experiments on the unidirectional recurrent VSR models. We apply our MV-based alignment to the existing models FRVSR[23], RLSP[8], and RSDN[11] to verify the effect of our MV-based alignment. In the original setting, FRVSR utilizes an optical flow to align the HR output from the past frame; RLSP and RSDN do not explicitly align the information from the previous frame. For a more comprehensive comparison, we also embed a pre-trained optical flow model SpyNet[21] into FRVSR, RLSP and RSDN to compare with our MV-based alignment. And we further fine-tune the SpyNet along with the model training. The training details follow the original works.

To evaluate the Residual informed sparse process, we first train a baseline recurrent VSR model without alignment. Then we apply MV-based alignment and Residual-based sparse processing to the baseline model to train our model. To balance model complexity and performance, the number of Resblocks for the recurrent module is set to 7. The number of feature channels is 128. We use Charbonnier loss[4] as pixel-wise loss since it better handles outliers and improves the performance over the conventional L2-loss[15]. The training details are provided in the supplementary material.

4.2 Effect of MV-based Alignment

We apply our MV-based alignment approach to the FRVSR, RLSP, and RSDN. The quantitative results are summarized in Tab. 1. XXX+Flow means that model XXX is aligned with the SpyNet. XXX+MV represents that model XXX is aligned with MVs. Original FRVSR aligns the HR estimation from the past frame by an optical flow model trained from scratch. In FRVSR+FLow, we replace the original optical flow model with pre-trained SpyNet and further refine the SpyNet when training. From the results, we can find FRVSR+Flow outperforms the original FRVSR. Probably because SpyNet estimates the optical flow more precisely than the original model. RLSP and RSDN do not explicitly align the information from the past frame. Due to the alignment, models with MV-based alignment achieve better performance than their original counterparts, even achieving comparable performance with the models with SpyNet. And we can see that as the CRF is increased, the performance gap between optical flow-based methods and MV-based methods narrows, which makes sense since when the CRF is large, the video compression artifacts are more apparent, and the optical flow estimate mistakes are more significant. So our MV-based alignment can replace the existing optical flow estimation model in unidirectional recurrent VSR models to save computation. For RLSP and RSDN, our approach can

Table 1: **The quantitative comparison (PSNR/ SSIM/ LPIPS)** on REDS4[27]. PSNR is calculated on Y-channel; SSIM and LPIPS are calculated on RGB-channel. Red and blue colors indicate the best and the second-best performance, respectively. 4× upsampling is performed.

Model	Compressed Results			Params (M)	Runtime (ms)
	CRF18	CRF23	CRF28		
FRVSR[23]	28.27/0.7367/0.3884	27.34/0.6965/0.4495	26.11/0.6492/0.5219	2.59	24
FRVSR+MV	29.01/0.7660/0.3470	27.77/0.7155/0.4141	26.32/0.6598/0.4969	0.84	20
FRVSR+Flow	29.15/0.7701/0.3393	27.85/0.7177/0.4076	26.32/0.6600/0.4928	2.28	32
RLSP[8]	28.46/0.7476/0.3614	27.47/0.7052/0.4243	26.20/0.6551/0.5015	4.37	27
RLSP+MV	29.26/0.7739/0.3309	27.95/0.7225/0.3973	26.43/0.6646/0.4815	4.37	28
RLSP+Flow	29.37/0.7769/0.3249	28.01/0.7242/0.3947	26.44/0.6651/0.4788	5.81	39
RSDN[11]	28.67/0.7575/0.3405	27.62/0.7144/0.3997	26.29/0.6642/0.4731	6.18	49
RSDN+MV	29.37/0.7804/0.3163	28.02/0.7294/0.3799	26.50/0.6724/0.4558	6.18	51
RSDN+Flow	29.59/0.7862/0.3094	28.13/0.7314/0.3770	26.51/0.6739/0.4523	7.62	62

achieve better performance with a tiny increase in runtime because of feature warping. It should be noted that our MV-based alignment does not increase the number of parameters. For FRVSR, because we remove its optical flow sub-model, our MV-based alignment can reduce the parameters and runtime but achieve superior performance over the original version.
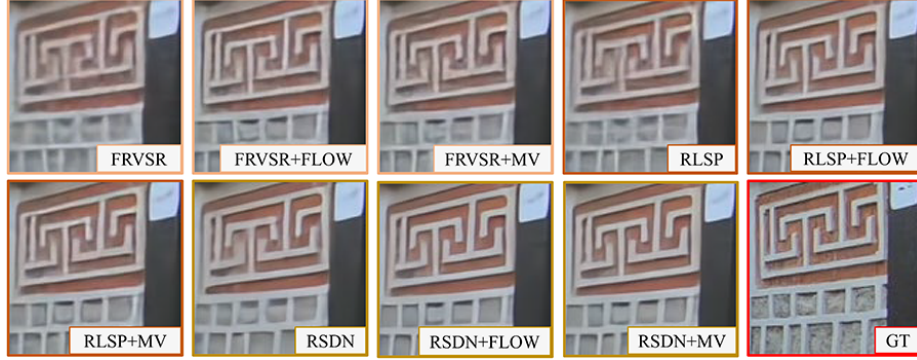

Fig. 4: Visual results on REDS4[27]

Fig. 4 shows the qualitative comparison. The models with our MV-based alignment restore finer details than the original FRVSR, RLSP, and RSDN. Compared with the models with optical flow estimation, our MV-aligned models achieve comparable visual results. More examples are provided in the Section 2.1 of supplementary material.

Table 2: **The quantitative comparison (PSNR/ SSIM/ LPIPS)** between image alignment and feature alignment on REDS4[27]. PSNR is calculated on Y-channel; SSIM and LPIPS are calculated on RGB-channel. The best results are highlighted in bold.

Model	CRF18	CRF23	CRF28
(a)	28.59/0.7546/0.3420	27.56/0.7122/0.3999	26.26/0.6622/0.4719
(b)	**29.32/0.7783/0.3186**	**28.00/0.7273/0.3818**	**26.47/0.6706/0.4569**
(c)	29.11/0.7675/0.3294	27.83/0.7172/0.3957	26.33/0.6604/0.4787

Image Alignment Vs Feature Alignment: As mentioned above, spatial alignment plays an important role in the VSR systems. The existing works with alignment can be divided into two categories: image alignment and feature alignment. We conduct experiments to analyze each of the categories and explain our design considerations about alignment. We design a recurrent baseline without alignment (Model (a)) and its MV-aligned versions. Model (b) is the MV-aligned model in feature space. And we apply MV-alignment on the HR prediction of the past frame to build a Model (c) with image alignment. The results are summarized in Tab. 2. The models with alignment outperform the baseline model, which further demonstrates the importance of alignment. And we find Model (b) achieves better performance than Model (c), so the alignment in feature space is more effective than in pixel level. The reason is that MV is block-wise motion estimation, the warped images inevitably suffer from information distortion. But there is a certain degree of redundancy in feature space, and this phenomenon is alleviated. Besides, the features contain more high-frequency information than images.

4.3   Effect of Residual Informed Sparse Processing

We apply the Residual informed sparse processing to the aligned model to get a more efficient model. The quantitative results are summarized in Tab. 3. The Baseline represents the baseline mentioned in Section 4.1; Baseline+MV means the MV-aligned model. MV+Res is the Residual-informed sparse processing. The **Sparse rate** is the ratio of pixels skipped by the network to all pixels in the image. As Tab. 3 shows, benefit from MV-based alignment, Baseline+MV achieves significant gains over the Baseline. The most gratifying result is that our sparse processing with MV-alignment and Residuals achieves a superior or comparable performance over Baseline with lots of computation saved. For the default CRF 23 in FFmpeg, our model can save about 70% computation on REDS4 and Vid4. CRF 18 means that the encoded video is visually lossless. So it needs more Residuals to decrease the encoding error. The sparse processing can save about 50% computation under this condition and achieve better performance than Baseline. For CRF 28, the sparse processing can save much more computation because the Residuals are sparser, and the performance is still comparable with the Baseline.

Table 3: **The quantitative results (PSNR/ SSIM/ Sparse rate)** of Residual informed sparse model on REDS4[27] and Vid4[18]. PSNR is calculated on Y-channel; SSIM is calculated on RGB-channel. The Sparse rate is the ratio of pixels skipped by the network to all pixels in the image. Red and blue colors indicate the best and the second-best performance, respectively. $4\times$ upsampling is performed.

Model	REDS4[27]			Vid4[18]		
	CRF18	CRF23	CRF28	CRF18	CRF23	CRF28
Baseline	28.59/0.7546/0.	27.56/0.7122/0.	26.26/0.6622/0.	24.61/0.6668/0.	23.91/0.6135/0.	22.87/0.5429/0.
Baseline+MV	29.32/0.7783/0.	28.00/0.7273/0.	26.47/0.6706/0.	25.13/0.6990/0.	24.20/0.6355/0.	23.01/0.5557/0.
MV+Res	29.03/0.7639/0.56	27.72/0.7131/0.75	26.15/0.6516/0.89	25.02/0.6800/0.49	24.04/0.6132/0.72	22.81/0.5333/0.90

We conduct qualitative comparisons on datasets REDS4 and Vid4. The results are shown in Fig. 5. The Residual informed model achieves finer details than the Baseline. More examples are provided in the Section 2.2 of supplementary material.

Bicubic   Baseline   Baseline+MV   MV+Res   GT

Fig. 5: Visual results of the Residual informed sparse process on Vid4[18] and REDS4[27]

CNN-based Mask Vs Residual-based Mask: We use a light CNN to predict the spatial mask for our Residual informed sparse processing during training. And when testing, we directly extract the Residuals from compressed videos to generate the spatial mask. In this section, we analyze the characteristics of the CNN-predicted mask and Residual-generated mask. As Fig. 6 shows, we can quickly identify the contours of objects and locate the details and textures from CNN-based masks. The Residual-based masks focus on the errors between the recurrent frame and the MV-warped past frame. Because Residuals are more likely to appear in the areas with details, the highlights of Residual-based masks also follow the location of details. Besides, the CNN-based masks are more continuous than the Residual-based mask. We also present the performance of the
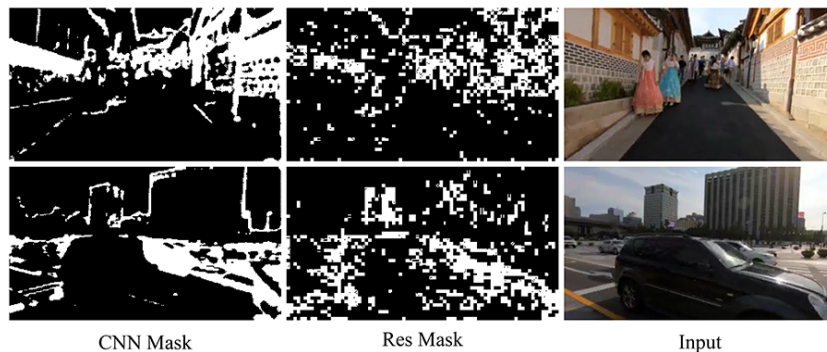
Fig. 6: Visual results of the spatial mask on REDS4[27]

models with CNN-based mask and Residual-based mask in Tab. 4. The results show that different from the Residual-based mask, the Sparse rate of CNN-base masks changes little with different CRF. So the CNN-based mask only highlights the main objects in the image. The Residual-based masks focus on the errors about MV-based alignment. For CRF 18, the information loss is slight, so the amount of Residuals is large, and the model achieves better performance than the model with the CNN-based mask. And for CRF 23 and 28, our model also outperforms the model with the CNN-based mask with a similar Sparse rate. The reason is that our Residual-based model follows the characteristics of video compression and is more suitable for models with MV-based alignment. Our Residual-based mask locates the "important" areas that need to be refined more precisely.

Table 4: **The quantitative comparison (PSNR/ SSIM/ Sparse rate)** about spatial mask on REDS4[27]. PSNR is calculated on Y-channel; SSIM and LPIPS are calculated on RGB-channel. The best results are highlighted in bold.

Model		CNN Mask	Res Mask
Compression results	CRF18	28.82/0.7492/**0.74**	**29.03/0.7639**/0.56
	CRF23	27.62/0.7040/**0.76**	**27.72/0.7131**/0.75
	CRF28	26.08/0.6456/0.79	**26.15/0.6516/0.89**

4.4 Temporal Consistency

Fig. 7 shows the temporal profile of the video super-resolution results, which is produced by extracting a horizontal row of pixels at the same position from consecutive frames and stacking them vertically. The "ResSparse Model" is the

model with our Residual informed sparse processing. The temporal profile produced by the model with our Residual informed sparse processing is temporally smoother, which means higher temporal consistency, and much sharper than the baseline model with about 70% computation of the baseline model saved when CRF is 23.
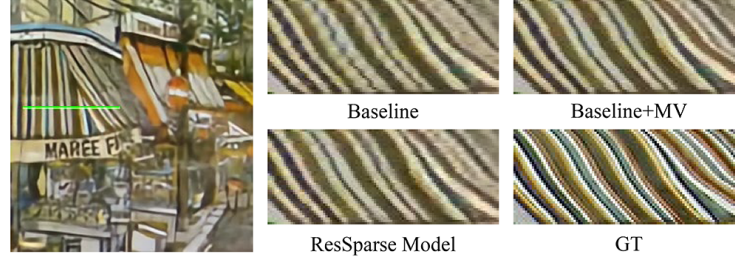

Fig. 7: Visualization of temporal profile for the green line on the calendar sequence with CRF 23.

5   Conclusion

This paper proposes to reuse codec information from compressed videos to assist the video super-resolution task. We employ Motion Vector to align mismatched frames in unidirectional recurrent VSR systems efficiently. Experiments have shown that Motion Vector based alignment can significantly improve performance with negligible additional computation. It even achieves comparable performance with optical flow based alignment. To further improve the efficiency of VSR models, we extract Residuals from compressed video and design Residual informed sparse processing. Combined with Motion Vector based alignment, our Residual informed processing can precisely locate the areas needed to calculate and skip the "unimportant" regions to save computation. And the performance of our sparse model is still comparable with the baseline. Additionally, given the importance of motion information for low-level video tasks and the inherent temporal redundancy of videos, our codec information assisted framework (CIAF) has the potential to be applied to other tasks such as compressed video enhancement and denoising.

References

1. Caballero, J., Ledig, C., Aitken, A.P., Acosta, A., Totz, J., Wang, Z., Shi, W.: Real-time video super-resolution with spatio-temporal networks and motion compensation. In: CVPR. pp. 2848–2857. IEEE Computer Society (2017)
2. Chan, K.C.K., Wang, X., Yu, K., Dong, C., Loy, C.C.: Basicvsr: The search for essential components in video super-resolution and beyond. In: CVPR. pp. 4947–4956. Computer Vision Foundation / IEEE (2021)
3. Chan, K.C.K., Zhou, S., Xu, X., Loy, C.C.: Basicvsr++: Improving video super-resolution with enhanced propagation and alignment. CoRR **abs/2104.13371** (2021)
4. Charbonnier, P., Blanc-Féraud, L., Aubert, G., Barlaud, M.: Two deterministic half-quadratic regularization algorithms for computed imaging. In: ICIP (2). pp. 168–172. IEEE Computer Society (1994)
5. Chen, P., Yang, W., Wang, M., Sun, L., Hu, K., Wang, S.: Compressed domain deep video super-resolution. IEEE Trans. Image Process. **30**, 7156–7169 (2021)
6. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: ICCV. pp. 764–773. IEEE Computer Society (2017)
7. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: ECCV (4). Lecture Notes in Computer Science, vol. 8692, pp. 184–199. Springer (2014)
8. Fuoli, D., Gu, S., Timofte, R.: Efficient video super-resolution through recurrent latent space propagation. In: ICCV Workshops. pp. 3476–3485. IEEE (2019)
9. Habibian, A., Abati, D., Cohen, T.S., Bejnordi, B.E.: Skip-convolutions for efficient video processing. In: CVPR. pp. 2695–2704. Computer Vision Foundation / IEEE (2021)
10. Haris, M., Shakhnarovich, G., Ukita, N.: Recurrent back-projection network for video super-resolution. In: CVPR. pp. 3897–3906. Computer Vision Foundation / IEEE (2019)
11. Isobe, T., Jia, X., Gu, S., Li, S., Wang, S., Tian, Q.: Video super-resolution with recurrent structure-detail network. In: ECCV (12). Lecture Notes in Computer Science, vol. 12357, pp. 645–660. Springer (2020)
12. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. In: ICLR (Poster). OpenReview.net (2017)
13. Jo, Y., Oh, S.W., Kang, J., Kim, S.J.: Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In: CVPR. pp. 3224–3232. Computer Vision Foundation / IEEE Computer Society (2018)
14. Kong, X., Zhao, H., Qiao, Y., Dong, C.: Classsr: A general framework to accelerate super-resolution networks by data characteristic. In: CVPR. pp. 12016–12025. Computer Vision Foundation / IEEE (2021)
15. Lai, W., Huang, J., Ahuja, N., Yang, M.: Deep laplacian pyramid networks for fast and accurate super-resolution. In: CVPR. pp. 5835–5843. IEEE Computer Society (2017)
16. Li, W., Tao, X., Guo, T., Qi, L., Lu, J., Jia, J.: Mucan: Multi-correspondence aggregation network for video super-resolution. In: ECCV (10). Lecture Notes in Computer Science, vol. 12355, pp. 335–351. Springer (2020)
17. Li, Y., Jin, P., Yang, F., Liu, C., Yang, M., Milanfar, P.: COMISR: compression-informed video super-resolution. CoRR **abs/2105.01237** (2021)
18. Liu, C., Sun, D.: A bayesian approach to adaptive video super resolution. In: CVPR. pp. 209–216. IEEE Computer Society (2011)

19. Liu, M., Zhang, Z., Hou, L., Zuo, W., Zhang, L.: Deep adaptive inference networks for single image super-resolution. In: ECCV Workshops (4). Lecture Notes in Computer Science, vol. 12538, pp. 131–148. Springer (2020)
20. Nah, S., Baik, S., Hong, S., Moon, G., Son, S., Timofte, R., Lee, K.M.: NTIRE 2019 challenge on video deblurring and super-resolution: Dataset and study. In: CVPR Workshops. pp. 1996–2005. Computer Vision Foundation / IEEE (2019)
21. Ranjan, A., Black, M.J.: Optical flow estimation using a spatial pyramid network. In: CVPR. pp. 2720–2729. IEEE Computer Society (2017)
22. Rec, B.I.: H.264, "advanced video coding for generic audiovisual services (2005)
23. Sajjadi, M.S.M., Vemulapalli, R., Brown, M.: Frame-recurrent video super-resolution. In: CVPR. pp. 6626–6634. Computer Vision Foundation / IEEE Computer Society (2018)
24. Sun, D., Yang, X., Liu, M., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: CVPR. pp. 8934–8943. Computer Vision Foundation / IEEE Computer Society (2018)
25. Tian, Y., Zhang, Y., Fu, Y., Xu, C.: TDAN: temporally-deformable alignment network for video super-resolution. In: CVPR. pp. 3357–3366. Computer Vision Foundation / IEEE (2020)
26. Wang, L., Dong, X., Wang, Y., Ying, X., Lin, Z., An, W., Guo, Y.: Learning sparse masks for efficient image super-resolution. CoRR **abs/2006.09603** (2020)
27. Wang, X., Chan, K.C.K., Yu, K., Dong, C., Loy, C.C.: EDVR: video restoration with enhanced deformable convolutional networks. In: CVPR Workshops. pp. 1954–1963. Computer Vision Foundation / IEEE (2019)
28. Wang, X., Yu, K., Dong, C., Loy, C.C.: Recovering realistic texture in image super-resolution by deep spatial feature transform. In: CVPR. pp. 606–615. Computer Vision Foundation / IEEE Computer Society (2018)
29. Wu, Z., Nagarajan, T., Kumar, A., Rennie, S., Davis, L.S., Grauman, K., Feris, R.S.: Blockdrop: Dynamic inference paths in residual networks. In: CVPR. pp. 8817–8826. Computer Vision Foundation / IEEE Computer Society (2018)
30. Xue, T., Chen, B., Wu, J., Wei, D., Freeman, W.T.: Video enhancement with task-oriented flow. Int. J. Comput. Vis. **127**(8), 1106–1125 (2019)
31. Yi, P., Wang, Z., Jiang, K., Jiang, J., Lu, T., Tian, X., Ma, J.: Omniscient video super-resolution. CoRR **abs/2103.15683** (2021)
32. Yi, P., Wang, Z., Jiang, K., Jiang, J., Ma, J.: Progressive fusion video super-resolution network via exploiting non-local spatio-temporal correlations. In: ICCV. pp. 3106–3115. IEEE (2019)
33. Zhang, Z., Sze, V.: FAST: A framework to accelerate super-resolution processing on compressed videos. In: CVPR Workshops. pp. 1015–1024. IEEE Computer Society (2017)
34. Zhu, X., Hu, H., Lin, S., Dai, J.: Deformable convnets V2: more deformable, better results. In: CVPR. pp. 9308–9316. Computer Vision Foundation / IEEE (2019)