Learning Spatiotemporal Frequency-Transformer for Compressed Video Super-Resolution

Zhongwei Qiu[1,2*], Huan Yang[3], Jianlong Fu[3], and Dongmei Fu[1,2]

[1] University of Science and Technology Beijing
[2] Shunde Graduate School of University of Science and Technology Beijing
[3] Microsoft Research
qiuzhongwei@xs.ustb.edu.cn, huayan@microsoft.com, jianf@microsoft.com, fdm_ustb@ustb.edu.cn

Abstract. Compressed video super-resolution (VSR) aims to restore high-resolution frames from compressed low-resolution counterparts. Most recent VSR approaches often enhance an input frame by "borrowing" relevant textures from neighboring video frames. Although some progress has been made, there are grand challenges to effectively extract and transfer high-quality textures from compressed videos where most frames are usually highly degraded. In this paper, we propose a novel Frequency-Transformer for compressed video super-resolution (FTVSR) that conducts self-attention over a joint space-time-frequency domain. First, we divide a video frame into patches, and transform each patch into DCT spectral maps in which each channel represents a frequency band. Such a design enables a fine-grained level self-attention on each frequency band, so that real visual texture can be distinguished from artifacts, and further utilized for video frame restoration. Second, we study different self-attention schemes, and discover that a "divided attention" which conducts a joint space-frequency attention before applying temporal attention on each frequency band, leads to the best video enhancement quality. Experimental results on two widely-used video super-resolution benchmarks show that FTVSR outperforms state-of-the-art approaches on both uncompressed and compressed videos with clear visual margins. Code are available at https://github.com/researchmm/FTVSR.

Keywords: VSR, Transformer, Frequency Learning, Compression

1 Introduction

Video super-resolution (VSR) aims to restore a sequence of high-resolution (HR) frames from its low-resolution (LR) counterparts. It is a fundamental computer vision task, and can benefit a broad range of downstream applications, such as video surveillance [39] and high-definition television [9]. State-of-the-art VSR approaches mainly focus on leveraging temporal information by sliding windows [14,17,29,30] or recurrent structures [2,27,37], and have achieved great success in limited scenarios that usually take uncompressed video frames as inputs.

[1] This work was done when Z. Qiu was an intern at Microsoft Research.

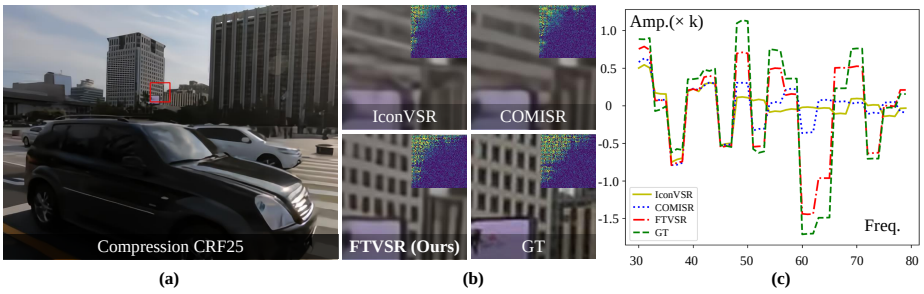

Fig. 1. Comparison of our FTVSR and state-of-the-art VSR methods (IconVSR [2], COMISR [20]) on compressed videos with a compression rate of CRF25. (a) The results of ×4 VSR by FTVSR. (b) Comparison of the zoom-in patches and their DCT-based spectral maps (shown in the top-right corner). FTVSR recovers more high-frequency information than IconVSR and COMISR. (c) Comparison of the Amplitude-Frequency curves on clipped frequency bands of 30 to 80. The proposed FTVSR is superior than other methods to approximate the curve of ground-truth (Best viewed in color)

However, most videos on the internet or in user devices are stored and transmitted in a compressed format. For example, the most widely-used video codec H.264 takes a constant rate factor (CRF) varied from 0 to 51 as its parameter to control the compression rate. As shown in Figure 1, directly applying the state-of-the-art IconVSR approach [2] to such a compressed scenario failed to generate visually pleasant results. Because the model trained on uncompressed videos often treats the unseen compression artifacts as common textures and magnifies these artifacts during restoration processes. Recent progress has been made by taking into account the compression artifacts in VSR model design. To strength the awareness of compression, a pioneer work, COMISR [20], is proposed to predict detail-aware flow to align high-resolution features and enhance HR images by Laplacian enhancement module. However, there are still large gaps between the generated frame and the ground-truth, as shown in Figure 1.

To solve the above issues, we propose a novel **F**requency **T**ransformer for compressed **V**ideo **S**uper-**R**esolution (**FTVSR**). The key insight is to transform a compressed video frame into a bunch of frequency-based patch representations by Discrete Cosine Transform (DCT), and design frequency-based attention to enable deep feature fusions across multiple frequency bands. Such a design has the following two key merits: 1) the DCT-based representation treats each frequency band "fairly", so that high-frequency visual details can be well-preserved; 2) the frequency attention enables low-frequency information (e.g., object structure) to guide the generation of high-frequency textures, so that the effect of compression artifacts can be greatly reduced. Besides, to further utilize the spatial and temporal dependencies in videos, we extensively explore different frequency attention mechanisms that combine with space and time attention in a proper manner. Extensive experiments on two widely-used VSR benchmarks demonstrate that the proposed FTVSR significantly outperforms previous meth-

ods and achieves new SOTA results. For example, for the setting of $CRF = 25$ in the REDS dataset, the gains of the proposed FTVSR are nearly 1.6 dB and 2.1 dB, compared with the competitive COMISR [20] and IconVSR [2], respectively.

2 Related Work

2.1 Video Super-Resolution

Uncompressed Video Super-Resolution Modern video super-resolution approaches [2,37,14,29,30,36,27,21] focus on improving the quality of HR sequences by extracting more information from temporal features, which can be categorized into sliding-window and recurrent structure. The approaches [14,29,30] based on sliding-window structure recover HR frames from adjacent LR frames within a sliding-window. They mainly use 3D convolution [14], optical flow [15,28] or deformable convolution [29,30] to align the temporal features. However, these methods can't utilize the temporal features from long-distance frames. Other approaches [28,11,12,37,2] based recurrent structure usually use a hidden state to transmit temporal information from long-distance frames. BasicVSR and IconVSR [2] achieve significant improvements with bidirectional recurrent structure, which fuses the forward and backward propagation features. Recently, transformer-based approaches [38,18,1] make great success by using different attention [8,40] to capture temporal features. Limited by computational costs, they just can aggregate information from a few adjacent frames. Despite the remarkable progress by these approaches, they are focus on uncompressed videos and usually fail to recover the HR frames from compressed LR frames.

Compressed Video Super-Resolution Compared with uncompressed VSR, compressed VSR is more difficult due to the lost information and the extra high-frequency artifacts caused by compression. There are three potential solutions to handle the compressed problem: video denoising, training on compressed videos and specific model design for compression. COMISR [20] firstly applies different video denoising [23,24,34] on compressed videos to remove the artifacts and uses state-of-the-art VSR methods [30,18,2] on the denoised LR videos. Experimental results have shown that this pre-process is not working since the degradation kernel used for training VSR is different from the denoising model. COMISR [20] further designs detail-aware module to align high-resolution features and Laplacian module to enhance HR frames with recurrent structure. However, these designs can not distinguish the high-frequency textures from the artifacts since these signals are coupled.

2.2 Frequency Learning

Lot of studies explore to learn in frequency domain, including high-level semantic tasks [6,33,26] and low-level restoration tasks [31,5,19,7]. High-level semantic

tasks usually reduce the computational cost by transforming images into frequency domain. Particularly, FcaNet [26] propose frequency channel attention to improve the performance of ResNet on classification task. Many low-level studies explore to restore content details from frequency decomposition perspective. Parts of them [7,19] study decomposing features into different frequency bands by multi-branch CNNs. Typically, OR-Net [19] uses multi-branch CNNs to separate different frequency components and enhances these features with frequency enhancement unit. Another parts [31,5] of them transform images into frequency domain. For example, D^3 [31] designs a dual-domain restoration network to remove artifacts of JPEG compressed images. Moreover, Ehrlich *et al.* [5] designs a Y-channel correction network and a color channel correction network in frequency domain to correct the JPEG artifact. Existing VSR methods are developed in pixel domain, but the video compression problem is generated in frequency domain. Inspired by this, we introduce a frequency-transformer to tackle the compression problems in VSR.

3 Approach

3.1 Problem formulation

VSR aims to restore the HR videos from its LR counterparts without taking into account video compression. Our focus, compressed VSR, aims to recover the HR frames from its compressed LR frames, which is more difficult. Let $I_{LR} = \{I_{LR}^t | t \in [1, T]\}$ be a compressed LR sequence of height H, width W, and frame length T. The restored super-resolution frames are denoted as $I_{SR} = \{I_{SR}^t | t \in [1, T]\}$ of height αH, width αW, in which α represents the upsampling scale factor. The corresponding HR frames are denoted as $I_{HR} = \{I_{HR}^t | t \in [1, T]\}$.

3.2 Frequency-based Tokenization

To solve the problem of compressed video super-resolution, we propose to adopt a frequency-based patch representation. Following the previous works [10,6,5] in computer vision, we adopt the widely-used method, DCT, as our operation to transfer an image into frequency domain.

DCT Discrete Cosine Transform projects an image into a set of cosine components for different 2D frequencies. Given an image patch P of height B and width B, a $B \times B$ DCT block D is generated as:

$$D(u, v) = c(u)c(v) \sum_{x=0}^{B-1} \sum_{y=0}^{B-1} P(x, y) cos[\frac{(2x+1)u\pi}{2B}] cos[\frac{(2y+1)v\pi}{2B}], \quad (1)$$

where x and y are the 2D indexes of pixels. $u \in [0, B-1]$ and $v \in [0, B-1]$ are the 2D indexes of frequencies. $c(\cdot)$ represents normalizing scale factor to enforce orthonormality and $c(u) = \sqrt{\frac{1}{B}}$ if $u = 0$, else $c(u) = \sqrt{\frac{2}{B}}$. The DCT and its inversion are denoted as DCT$(\cdot)$ and rDCT$(\cdot)$, respectively.

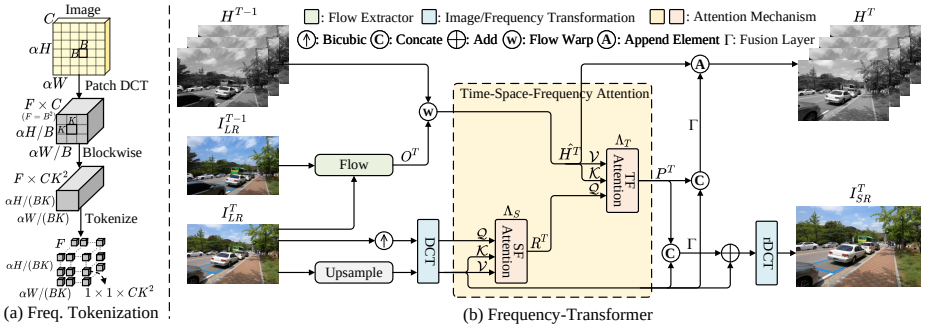

Fig. 2. (a) An RGB image is extracted as frequency tokens of size $C \times K \times K$ by DCT-based frequency tokenization. (b) A Frequency-Transformer with divide time-space-frequency (TSF) attention Λ_{ST}, which achieves best performance in our experiments. Given compressed LR sequence, Frequency-Transformer performs TSF attention on the frequency tokens of video frames and output SR frames with a hidden state H maintained by a recurrent structure. TSF consists of Λ_S attention and Λ_T attention. The Q, K, V of Λ_S are tokens from videos frame sampled I^T_{LR} by bicubic and upsample network, respectively. R^T is the output of Λ_S, further sums with hidden states $\hat{H}^T$ warped from past hidden states by flow O^T, as the K and V of Λ_T attention. P^T is the output of Λ_T, which further used to update hidden state and recover SR frame I^T_{SR}

DCT-based Frequency Tokenization Given a LR sequence, we firstly upsample the I_{LR} by a upsampling network $\varphi(\cdot)$. For each frame, we transform each channel of RGB image into frequency domain by applying DCT on the patches of shape $B \times B$ as Equation 1, which can be formulated as:

$$D_{LR}(u, v) = \mathrm{DCT}(\varphi(I_{LR})), \tag{2}$$

where $D_{LR}(u, v)$ of shape $T \times F \times C \times \frac{\alpha H}{B} \times \frac{\alpha W}{B}$ represents the transformed 2D spectral map from LR image. T, F, C, $\frac{\alpha H}{B}$ and $\frac{\alpha W}{B}$ represent sequence length, frequency dimensions, image channels, height and width, respectively. The frequency number is $F = B^2$.

For a spectral frame $D_{LR}(u, v)$, we split the frequency dimension to form F visual tokens. The frequency tokens set $\mathcal{T}$ can be represented as:

$$\mathcal{T} = \{\tau_f, f \in [1, F]\}, \tag{3}$$

where τ_f represents the frequency token in f^{th} frequency, which has a feature size of $C \times \frac{\alpha H}{B} \times \frac{\alpha W}{B}$. This frequency tokenization mechanism brings the information exchange between different frequency bands and forces neural network treating low-frequency signals and high-frequency signals "fairly", which is beneficial to preserve high-frequency visual details. Combined with frequency attention mechanism in Section 3.3, the high-frequency textures can be restored well by the guidance of low-frequency information (e.g., object structure).

In order to capture the frequency relationship between different spatial blocks, the spectral maps are split into a set of blocks with a kernel size of $K \times K$. To

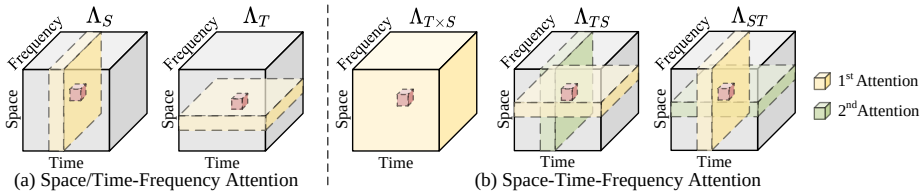

Fig. 3. The visualization of (a) space/time-frequency attention, and (b) space-time-frequency attention. The red cube denotes the query token. The yellow area and green area represent the candidate area for computing attention with query following the order of yellow first and then green

further extract temporal information, we extend the same tokenization to all video frames. Therefore, we generate more fine-grained frequency tokens in both space and time dimensions, which can be represented as:

$$\mathcal{T} = \{\tau_{(t,i,f)}, t \in [1,T], i \in [1,N], f \in [1,F]\}, \tag{4}$$

where each token $\tau_{(t,i,f)}$ has a shape of $C \times K \times K$. N represents the generated block number in each frame. Different from traditional vision Transformers [4,22,1], which crop image patches and form a set of spatial visual tokens, our tokens are based on different frequency bands. In a nutshell, we generate N blocks for each spectral frame D^t_{LR}, and each block has DCT-based frequency tokens τ of the number of F. The total number of frequency tokens is $T \times N \times F$. Figure 2 (a) presents more details about the whole tokenization process.

3.3 Frequency-based Attention

The inputs of frequency transformer are DCT-based visual tokens, which have been generated in Section 3.2. To better take advantage of temporal information for VSR, the query tokens $\mathcal{Q}$ are extracted from spectral map D^T_{LR}. Keys $\mathcal{K}$ and values $\mathcal{V}$ are extracted from spectral maps $\{D^t_{LR}, t \in [1, T-1]\}$. For a target frame D^T_{LR}, the query, key, and value sets are denoted as:

$$\begin{aligned}
\mathcal{Q} &= \{\tau^q_{(T,i,f)}, i \in [1,N], f \in [1,F]\}, \\
\mathcal{K} &= \{\tau^k_{(t,i,f)}, t \in [1,T-1], i \in [1,N], f \in [1,F]\}, \\
\mathcal{V} &= \{\tau^v_{(t,i,f)}, t \in [1,T-1], i \in [1,N], f \in [1,F]\},
\end{aligned} \tag{5}$$

where $\tau^q_{(T,i,f)}$, $\tau^k_{(t,i,f)}$, and $\tau^v_{(t,i,f)}$ represent the query, key, and value tokens, respectively. Each token is extracted from spectral maps among time, space, and frequency dimensions according to needs of computing different kinds of frequency attention, which will be discussed as follows.

Frequency Attention The frequency attention aims to capture the relationship between different frequency bands. Given a query token τ^q_f at the f^{th} fre-

quency, the uniform formulation of frequency attention (denoted as Λ) is:

$$\Lambda(\tau_f^q, \tau_{\hat{f}}^k, \tau_{\hat{f}}^v) = \mathrm{SM}(\frac{\tau_f^q \cdot \tau_{\hat{f}}^k}{\sqrt{d^k}})\tau_{\hat{f}}^v, \hat{f} \in [1, F], \tag{6}$$

where SM represents the softmax activation function and d^k denotes the normalization factor. Note that there is a feed forward network (FFN) after frequency attention, which is omitted in this paper. However, computing frequency attention on whole spectral maps is impractical since the feature size of spectral map is different during the process of training and inference. Therefore, we adopt the way of computing frequency attention on spatial blocks of spectral maps. To explore different frequency attention mechanisms combined with time or space attention. As shown in Figure 3, we propose space-frequency attention, time-frequency attention, and time-space-frequency attention.

Space/Time-Frequency Attention Space-Frequency (SF) attention computes the frequency attention weights between spatial blocks. The visualization of SF is shown in Figure 3 (a). For a query token $\tau_{(i,f)}^q$ at the f^{th} frequency in the i^{th} block, the SF attention is $\Lambda_S(\tau_{(i,f)}^q, \tau_{(\hat{i},\hat{f})}^k, \tau_{(\hat{i},\hat{f})}^v), \hat{i} \in [1, N], \hat{f} \in [1, F]$, which computes the frequency attention as Equation 6 in spatial dimension. The inputs of Λ_S are space-frequency tokens $\tau_{(i,f)}$. Since the tokens are extracted from both space and frequency dimensions, $N \times F$ tokens are generated for SF attention.

The Time-Frequency (TF) attention is computed on the blocks with the same spatial position from different video frames. The visualization of TF attention is shown in Figure 3 (a). Given a query token $\tau_{(t,f)}^q$, the TF attention is $\Lambda_T(\tau_{(t,f)}^q, \tau_{(\hat{t},\hat{f})}^k, \tau_{(\hat{t},\hat{f})}^v), \hat{t} \in [1, T-1], \hat{f} \in [1, F]$, which computes frequency attention as Equation 6 in temporal dimension. The inputs of Λ_T are time-frequency tokens $\tau_{(t,f)}$. Since the tokens are extracted from both time and frequency dimensions, $T \times F$ tokens are generated for TF attention.

Time-Space-Frequency Attention Both the temporal and spatial information are important for compressed VSR. To further explore the frequency attention in both spatial and temporal dimensions, we propose Time-Space-Frequency (TSF) attention. TSF are the combinations of SF and TF attention. It can be divided into two types: joint SF and TF attention, divided SF and TF attention. The visualizations of TSF are shown in Figure 3 (b). Given a query token $\tau_{(t,i,f)}^q$, joint TSF attention is $\Lambda_{T\times S}(\tau_{(t,i,f)}^q, \tau_{(\hat{t},\hat{i},\hat{f})}^k, \tau_{(\hat{t},\hat{i},\hat{f})}^v)$, $\hat{t} \in [1, T-1], \hat{i} \in [1, N], \hat{f} \in [1, F]$, which computes the frequency attention as Equation 6 in both spatial and temporal dimensions. The inputs of joint TSF attention are time-space-frequency tokens $\tau_{(t,i,f)}$. Since the tokens are extracted among time, space, and frequency dimensions, $T \times N \times F$ tokens are generated for joint TSF.

For divided TSF attention, two types of TSF are designed according to the order of computing TF and SF attention. One of them can be formulated as:

$$\begin{aligned} &\Lambda_{ST}(\tau^q_{(t,i,f)}, \tau^k_{(\hat{t},\hat{i},\hat{f})}, \tau^v_{(\hat{t},\hat{i},\hat{f})}) = \Lambda_T(\hat{\tau}^q_{(t,f)}, \tau_{(\hat{t},\hat{f})}, \tau_{(\hat{t},\hat{f})}), \\ &where\ \hat{\tau} = \Lambda_S(\tau^q_{(i,f)}, \tau^k_{(\hat{i},\hat{f})}, \tau^k_{(\hat{i},\hat{f})}), \hat{t} \in [1, T-1], \hat{i} \in [1, N], \hat{f} \in [1, F]. \end{aligned} \tag{7}$$

The divided TSF attention Λ_{ST} represents the attention that computes space-frequency attention Λ_S firstly, then computes time-frequency attention Λ_T. In our experiments, Λ_{ST} performs best for frequency transformer. This is because in compressed VSR, degraded frames should be first restored by the space-frequency attention then the recovered textures could be used to benefit temporal learning in the time-frequency attention.

The other one can be formulated as $\Lambda_{TS}(\tau^q_{(t,i,f)}, \tau^k_{(\hat{t},\hat{i},\hat{f})}, \tau^v_{(\hat{t},\hat{i},\hat{f})})$. The divided TSF attention Λ_{TS} represents the attention that computes time-frequency attention Λ_T firstly, then computes space-frequency attention Λ_S. The computing process of Λ_{TS} is similar with Equation 7.

3.4 Frequency Transformer

To recover HR sequences, we use the similar recurrent structure as TTVSR [21]. Each HR frame is restored from its LR counterparts and a propagation hidden state H. Given a LR frame I^T_{LR}, the SR frame can be restored as:

$$\begin{aligned} I^T_{SR} &= \text{rDCT}(T_{freq}(\mathcal{Q}, \mathcal{K}, \mathcal{V})) \\ &= \text{rDCT}(\Gamma(A_{freq}(\mathcal{Q}, \mathcal{K}, \mathcal{V}), D^T_{LR}) + D^T_{LR}), \end{aligned} \tag{8}$$

where T_{freq} represents the Frequency Transformer. A_{freq} represents the frequency attention used in T_{freq} and $A_{freq} \in \{\Lambda_S, \Lambda_T, \Lambda_{T\times S}, \Lambda_{TS}, \Lambda_{ST}\}$. Γ represents the fusion operation which concatenates the outputs of A_{freq} and D^T_{LR}, then reduces the dimensions of the concatenated features by Linear layer.

For example, a frequency Transformer formed by divided TSF attention Λ_{SF} is shown in Figure 2 (b). The output P^T of TSF can be formulated as:

$$\begin{aligned} P^T &= \Lambda_T(R_T, \hat{H}^T, \hat{H}^T), \\ where\ R^T &= \Lambda_S(\mathcal{Q}_S, \mathcal{K}_S, \mathcal{V}_S), \hat{H}^T = W(H^{T-1}, O^T). \end{aligned} \tag{9}$$

$\hat{H}^T$ represents the hidden states warped from past frames H^{T-1} according to flow O^T. W represents the flow warp operation as [2]. H^T is updated by the output P^T of TSF attention and the DCT-based features D^T_{LR}. $\mathcal{Q}_S$ are extracted from upsampled I^T_{LR} by Bicubic upsampling while $\mathcal{K}$ and $\mathcal{V}$ are extracted from upsampled I^T_{LR} by a upsample neural network. The difference between upsample operations brings the location guidance of the hard-to-recover parts, which should pay more attention to it. $\mathcal{Q}_T$ is the temporal-frequency query tokens for Λ_T. The output P^T of TSF attention Λ_{SF} is used to recover SR frames, which can be formulated as:

$$I^T_{SR} = \text{rDCT}(\Gamma(P^T, D^T_{LR}) + D^T_{LR}). \tag{10}$$

More details about the network structure of our proposed Frequency-Transformer can be found in the supplementary material.

We follow the previous works [2,20], using Charbonnier penalty loss [16], which is applied on each video frames. The total loss $\mathcal{L}$ is the average of frames,

$$\mathcal{L} = \frac{1}{T}\sum_{t=1}^{T}\sqrt{||I_{HR}^{t} - I_{SR}^{t}||^2 + \epsilon^2}, \tag{11}$$

where ϵ is a constant value and $\epsilon = 1e-3$.

4 Experiments

4.1 Implementation Details

During training, the Cosine Annealing scheme and Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.99$ are used. The initial learning rate of FTVSR is 2×10^4. The batch size is 8 videos. The training frame length is 40 for final results and 10 for ablation study. The input patch size is 64×64 and the SR scale is $4\times$. Data augmentations include random horizontal flips, vertical flips, and rotations. We train FTVSR with 400k iterations for the final model and 100k iterations for quick ablation study. All ablation study are based on the backbone of BasicVSR [2] and final model is based on the backbone of TTVSR [21] for better results. Unless otherwise stated, FTVSR is trained with a ratio of 50% uncompressed videos and 50% compressed videos. The compressed videos are uniformly sampled from different compression rates. During inference, we pad the input images with the edge values to keep they can be transformed into spectral maps by DCT and remove the padding after transforming spectral maps into pixel images by rDCT. We crop images into 4×4 patches for inference since the limitation of CUDA memory.

4.2 Datasets and Evaluation Metrics

Datasets Following the previous works [1,2], we use REDS [25] and Vimeo-90K [35] for training. The REDS dataset contains 270 videos and each video in has 100 frames with a resolution of 1280×720. For a fair comparison, four sequences as previous works [2,1,30,18] for testing, called REDS4. The Vimeo-90K contains 64,612 sequences for training. Each video contains 7 frames with a resolution of 448×256. Same as previous works [20,2], the testing set of Vimeo-90K is Vid4, which contains four videos. Each video includes 30 to 50 frames.

Evaluation Metrics We use the same metrics peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) [32] as previous works [2,1,30,18] in our evaluation. In addition, for compression videos, we use the most common setting for H.264 codec at different compression rates (different CRF values). Following previous COMISR [20], we use CRF of 15, 25, and 35 to generate compressed

Table 1. Quantitative comparison on the **compressed** videos of REDS4 [25] for 4× VSR. Each entry shows the PSNR↑/SSIM↑ on RGB channels as [2,20]. Red indicates the best and blue indicates the second best performance (Best viewed in color)

Method	Per clip with Compression CRF25				Average of clips with Compression		
	Clip_000	Clip_011	Clip_015	Clip_020	CRF15	CRF25	CRF35
DUF [13]	23.46/0.622	24.02/0.686	25.76/0.773	23.54/0.689	25.61/0.775	24.19/0.692	22.17/0.588
FRVSR [27]	24.25/0.631	25.65/0.687	28.17/0.770	24.79/0.694	27.61/0.784	25.72/0.696	23.22/0.579
EDVR [30]	24.38/0.629	26.01/0.702	28.30/0.783	25.21/0.708	28.72/0.805	25.98/0.706	23.36/0.600
TecoGan [3]	24.01/0.624	25.39/0.682	27.95/0.768	24.48/0.686	26.93/0.768	25.46/0.690	22.95/0.589
RSDN [12]	24.04/0.602	25.40/0.673	27.93/0.766	24.54/0.676	27.66/0.768	25.48/0.679	23.03/0.579
MuCAN [18]	24.39/0.628	26.02/0.702	28.25/0.781	25.17/0.707	28.67/0.804	25.96/0.705	23.55/0.600
BasicVSR [2]	24.37/0.628	26.01/0.702	28.13/0.777	25.21/0.709	29.05/0.814	25.93/0.704	23.22/0.596
IconVSR [2]	24.35/0.627	26.00/0.702	28.16/0.777	25.22/0.709	29.10/0.816	25.93/0.704	23.22/0.596
COMISR [20]	24.76/0.660	26.54/0.722	29.14/0.805	25.44/0.724	28.40/0.809	26.47/0.728	23.56/0.599
FTVSR	26.06/0.703	28.71/0.779	30.17/0.839	27.26/0.782	30.51/0.853	28.05/0.776	24.82/0.657

videos. Detailed command for video compression can be found in the supplementary material. We then evaluate FTVSR and report the PSNR and SSIM on these compressed videos with these CRF values.

4.3 Comparison with State-of-the-art Methods

Evaluation on Compressed Videos We compare FTVSR with other state-of-the-art methods on REDS [25] and Vid4 [35] datasets. Following the compressed settings as COMISR [20], we compress the videos with several compression rates (CRF15, CRF25, CRF35) and evaluate on the compressed videos in PSNR and SSIM.

For REDS [25] dataset, the results on compressed videos are shown in Table 1 and results of other methods are cited from [20]. For BasicVSR and IconVSR, we finetune them on the compressed videos as the same training settings of [20]. Although recent BasicVSR [2] and IconVSR [2] achieve state-of-the-art results on uncompressed videos, they perform not well on the compressed videos. For example, BasicVSR achieves 25.93dB and 23.22dB in PSNR of compression CRF25 and CRF35. Besides, IconVSR, which performs better than BasicVSR on uncompressed videos, but just obtain 25.93dB and 23.22dB in PSNR of compression CRF25 and CRF35 same as the BasicVSR. This phenomenon indicates that only increases the model capacity has less effect on compression problems.

COMISR [20] alleviates the compression problem to some extent by its special designs for compression, but the gains are small (e.g., 26.47dB and 23.56dB in PSNR with a compression rate of CRF25 and CRF35). However, FTVSR achieves 30.51, 28.05dB, and 24.82dB in PSNR on compressed videos with a compression rate of CRF15, CRF25, and CRF35, respectively. FTVSR outperforms SOTA COMISR by 1.6dB on the compressed videos in CRF25. The results show that FTVSR has strong capabilities on compression problems.

For Vid4 [35] dataset, the results on compressed videos are shown in Table 2. The results of BasicVSR and IconVSR are obtained by finetuning on compressed

Table 2. Quantitative comparison on the **compressed** video of Vid4 [35] for 4× VSR. Following previous works [2,20], each entry shows the PSNR↑/SSIM↑ on Y-channel. Red and blue indicates the best and second best performances (Best viewed in color)

Method	Per clip with Compression CRF25				Average of clips with Compression		
	calendar	city	foliage	walk	CRF15	CRF25	CRF35
DUF [13]	21.16/0.634	23.78/0.632	22.97/0.603	24.33/0.771	24.40/0.773	23.06/0.660	21.27/0.515
FRVSR [27]	21.55/0.631	25.40/0.575	24.11/0.625	26.21/0.764	26.01/0.766	24.33/0.655	22.05/0.482
EDVR [30]	21.69/0.648	25.51/0.626	24.01/0.606	26.72/0.786	26.34/0.771	24.45/0.667	22.31/0.534
TecoGan [3]	21.34/0.624	25.26/0.561	23.50/0.592	25.73/0.756	25.25/0.741	23.94/0.639	21.99/0.479
RSDN [12]	21.72/0.650	25.28/0.615	23.69/0.591	25.57/0.747	26.58/0.781	24.06/0.650	21.29/0.483
MuCAN [18]	21.60/0.643	25.38/0.620	23.93/0.599	26.43/0.782	25.85/0.753	24.34/0.661	22.26/0.531
BasicVSR [2]	21.64/0.641	25.45/0.620	23.79/0.586	26.26/0.774	26.56/0.780	24.28/0.656	21.97/0.509
IconVSR [2]	21.67/0.644	25.46/0.621	23.83/0.588	26.26/0.774	26.65/0.782	24.31/0.657	21.97/0.509
COMISR [20]	22.81/0.695	25.94/0.640	24.66/0.656	26.95/0.799	26.43/0.791	24.97/0.701	22.35/0.509
FTVSR	22.97/0.720	26.29/0.670	24.94/0.664	27.30/0.816	27.40/0.811	25.38/0.706	22.61/0.540

Table 3. Evaluation on the **uncompressed** videos of REDS4 [25] and Vid4 [35] for 4× VSR. Each entry shows the PSNR↑/SSIM↑. ∗ represents the FTVSR is trained on only uncompressed videos. † represents FTVSR is trained on both compressed and uncompressed videos, which is a more difficult setting. All other methods are trained on uncompressed videos and evaluated on uncompressed videos

Datasets	TOFlow[35]	DUF[13]	EDVR[30]	COMISR[20]	BasicVSR[2]	IconVSR[2]	**FTVSR***	**FTVSR**†
REDS4	27.98/0.799	28.63/0.825	31.09/0.880	29.68/0.868	31.42/0.890	31.67/0.895	31.82/0.896	31.74/0.895
Vid4	25.85/0.766	27.38/0.832	27.85/0.850	27.31/0.840	27.96/0.855	28.04/0.857	28.31/0.860	28.06/0.856

videos as [20]. For a fair comparison, we also adopt the same compression settings as COMISR [20]. On compressed videos with a compression rate of CRF 15, 25, and 35, FTVSR achieves 27.40dB, 25.38dB, and 22.61dB in PSNR, respectively. FTVSR outperforms other methods. These results demonstrate the huge potential of FTVSR on the task of compressed VSR.

We also visualize the results of FTVSR and SOTA methods on compressed videos. As shown in Figure 4, FTVSR performs well on both compressed and uncompressed video. Especially on the compressed video with CRF25 and CRF35, the visual quality of FTVSR is superior to other methods. An interesting phenomenon is that COMISR performs better than BasicVSR and IconVSR on compressed videos, but poorly on uncompressed videos. However, our FTVSR performs well on both compressed videos and uncompressed videos as shown in Figure 4. Especially on the cases with a compression rate of CRF35, BasicVSR, IconVSR, and COMISR are failed to recover the texture, but FTVSR still performs well on these cases. It's because frequency attention enables low-frequency information to guide the generation of high-frequency textures.

Evaluation on Uncompressed Videos To study the potential of FTVSR, we also evaluate FTVSR on the uncompressed videos of the REDS and Vid4 datasets, respectively. For a fair comparison, we compare with SOTA methods in two settings: 1) FTVSR*, training only on uncompressed videos, and testing on uncompressed videos. 2) FTVSR†, training on both compressed and uncom-

Table 4. Comparison of parameters, FLOPs and PSNR↑/SSIM↑ on the compressed videos with CRF25. FLOPs is computed on one LR frame with the size of 180×320 and $\times 4$ upsampling on the REDS4 dataset

Methods	DUF[13]	EDVR[30]	MuCAN[18]	BasicVSR[2]	IconVSR[2]	COMISR[20]	FTVSR
Params(M)	5.8	20.6	13.6	6.3	8.7	6.2	10.8
FLOPs(T)	2.34	2.95	>1.07	0.33	0.51	0.36	0.76
PSNR/SSIM	24.19/0.692	25.98/0.706	25.96/0.705	25.93/0.704	25.93/0.704	26.47/0.728	27.28/0.763

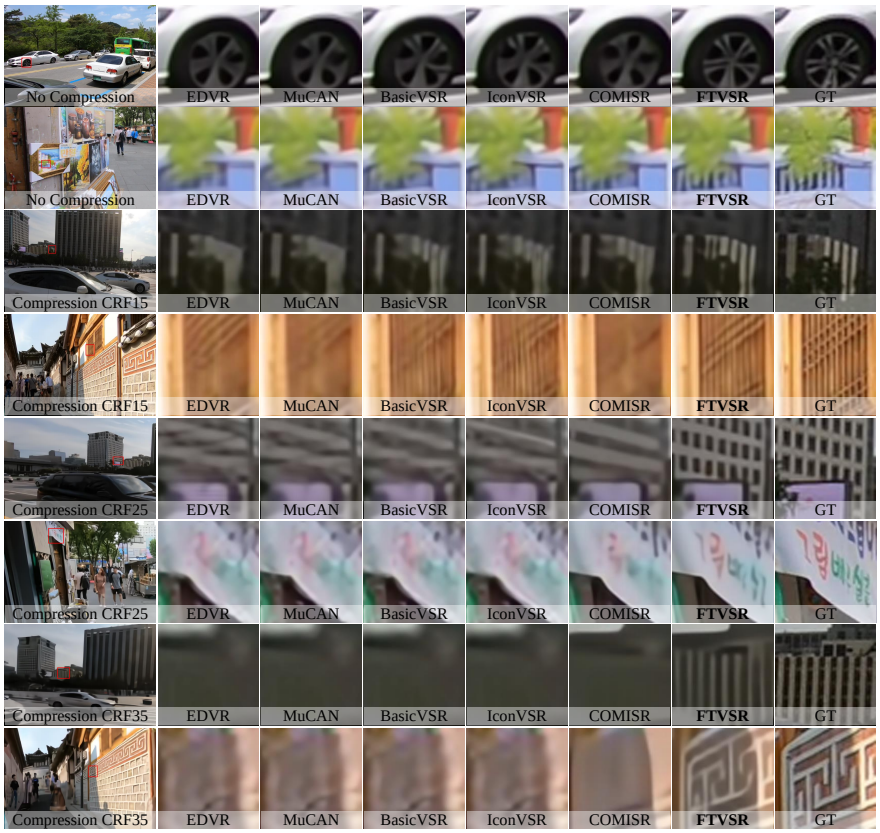

Fig. 4. Visualization results of our FTVSR and other VSR methods on the uncompressed videos and compressed videos with compression rates of CRF 15, 25, and 35

pressed videos, and testing on uncompressed videos, which is a more difficult setting for evaluating on uncompressed videos since the compressed data brings more noises for VSR model. As shown in Table 3, all results are evaluated on uncompressed videos. The results of other methods are all obtained from their paper and their model is trained on only uncompressed videos as setting 1. For the setting 1 which is fair for our method, FTVSR* outperforms SOTA IconVSR [2] in PSNR on both REDS4 and Vid4 datasets. Moreover, for setting

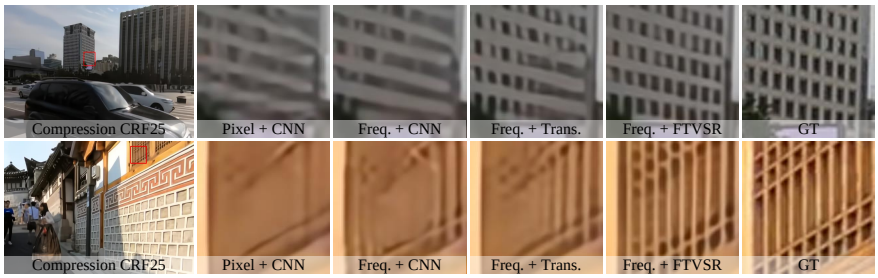

Fig. 5. Visualization results of "Pixel + CNN", "Frequency + CNN", "Frequency + Transformer", and "Frequency + FTVSR" in Table 5

Table 5. Ablation study of FTVSR (PSNR↑/SSIM↑) on the REDS4 dataset

Domain + Backbone	Per clip with Compression CRF25				Average of clips with Compression		
	Clip_000	Clip_011	Clip_015	Clip_020	CRF15	CRF25	CRF35
Pixel + CNN	24.37/0.628	26.01/0.702	28.13/0.777	25.21/0.709	29.05/0.814	25.93/0.704	23.22/0.596
Frequency + CNN	24.98/0.666	27.11/0.746	29.36/0.818	26.05/0.751	29.20/0.825	26.87/0.745	23.83/0.629
Frequency + Transformer	25.20/0.684	27.53/0.763	29.47/0.828	26.33/0.766	29.51/0.837	27.15/0.759	24.03/0.644
Frequency + FTVSR	25.26/0.609	27.75/0.766	29.62/0.831	26.47/0.772	29.70/0.843	27.28/0.763	24.22/0.646

Table 6. Comparisons of different types of frequency attention on the compressed videos of REDS with compression rates of CRF 15, 25, and 35. All the methods in this table are in the frequency domain. "Base" represents the traditional attention without frequency attention mechanism. Each entry shows PSNR↑/SSIM↑

Attention	Base	Λ_S	Λ_T	$\Lambda_{T\times S}$	Λ_{TS}	Λ_{ST}
CRF15	29.51/0.837	29.63/0.840	29.60/0.840	29.61/0.839	29.65/0.841	**29.70/0.843**
CRF25	27.15/0.759	27.23/0.761	27.11/0.760	27.22/0.760	27.24/0.762	**27.28/0.763**
CRF35	24.03/0.644	24.12/0.646	24.05/0.641	24.11/0.644	24.12/0.645	**24.22/0.646**

2, FTVSR† achieves 31.74dB in PSNR on REDS4 dataset, which outperforms IconVSR [2] although IconVSR is trained on full clean uncompressed videos. Besides, FTVSR† obtain comparable results on Vid4 dataset. Compared with COMISR [20] that performs well on compressed videos while unsatisfactory on uncompressed videos, FTVSR performs well on both compressed videos and uncompressed videos, even in the difficult setting of that the model is trained on both compressed videos and uncompressed videos.

Comparison of Parameters and FLOPs The comparisons of parameters and FLOPs are shown in Table 4. The FLOPs are computed with the input of LR frame size 180×320 and conducting $4\times$ upsampling VSR task. Based on BasicVSR, FTVSR outperforms other methods with the comparable parameters and FLOPs. FTVSR adopt a similar architecture as BasicVSR and process HR frames in the frequency domain. The FLOPs are comparable with BasicVSR since the DCT operation reduces the computational costs of FTVSR.

4.4 Ablation Study

To evaluate the effectiveness of FTVSR, we conduct the ablation study on the REDS4 dataset. As shown in Table 5, We use BasicVSR [2] as baseline, which learns in pixel domain and achieves 29.05dB, 25.93dB, 23.22dB in PSNR on the compressed videos with compression CRF15, CRF25, and CRF35, respectively. The performances are poor compared with its 31.42dB on uncompressed videos. Then, we transfer images into the frequency domain, which obtains a relative gain of 0.94dB in PSNR on the compressed videos with CRF25. In the frequency domain, a transformer-based model without frequency attention achieves 27.15dB in PSNR on the compressed videos with CRF25, which shows that the attention mechanism is beneficial for frequency learning. Replacing the basic transformer by our FTVSR, FTVSR achieves 27.28dB in PSNR on the compressed videos with CRF25, which shows that the frequency attention is better than traditional attention in the frequency domain. As shown in Figure 5, FTVSR achieves better visualization results than others.

To evaluate the effectiveness of different frequency attention introduced in Section 3.3, we conduct the ablation study on the REDS dataset. As shown in Table 6,"Base" represents a traditional transformer which computes spatial attention in frequency domain. The results of base attention are lower than frequency attention. For the different frequency attentions, space-frequency attention, time-frequency attention, joint time-space-frequency attention and divided time-space-frequency attention ($\{\Lambda_S, \Lambda_T, \Lambda_{T\times S}, \Lambda_{TS}, \Lambda_{ST}\}$), the results in Table 6 show that the divided frequency attention (Λ_{ST}) with an order of space first and time later is better. This is because in compressed VSR, degraded frames should be first restored by the space-frequency attention then the recovered textures could be used to benefit temporal learning in the time-frequency attention.

5 Conclusions

In this paper, we propose a novel spatiotemporal Frequency-Transformer for compressed Video Super-Resolution (FTVSR). To handle the compression issues, we transform compressed video frames into frequency domain and design frequency-based attention to enable the feature fusions across multiple frequency bands. The frequency-based tokenization and frequency attention mechanism enables low-frequency information to guide the generation of high-frequency textures. To utilize spatial and temporal information, we further explore the different types of frequency attention combined with space and time attentions. Experiments on two widely-used VSR datasets show that the proposed FTVSR significantly outperforms previous works and achieves new SOTA results.

Acknowledgments

This work was supported by the Scientific and Technological Innovation of Shunde Graduate School of University of Science and Technology Beijing (No. BK20AE004 and No.BK19CE017).

References

1. Cao, J., Li, Y., Zhang, K., Van Gool, L.: Video super-resolution transformer. arXiv preprint arXiv:2106.06847 (2021)
2. Chan, K.C., 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 (2021)
3. Chu, M., Xie, Y., Mayer, J., Leal-Taixé, L., Thuerey, N.: Learning temporal coherence via self-supervision for gan-based video generation. ACM TOG **39**(4), 75–1 (2020)
4. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
5. Ehrlich, M., Davis, L., Lim, S.N., Shrivastava, A.: Quantization guided jpeg artifact correction. In: ECCV. pp. 293–309. Springer (2020)
6. Ehrlich, M., Davis, L.S.: Deep residual learning in the jpeg transform domain. In: ICCV. pp. 3484–3493 (2019)
7. Fritsche, M., Gu, S., Timofte, R.: Frequency separation for real-world super-resolution. In: ICCVW. pp. 3599–3608. IEEE (2019)
8. Fu, J., Zheng, H., Mei, T.: Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In: CVPR. pp. 4438–4446 (2017)
9. Goto, T., Fukuoka, T., Nagashima, F., Hirano, S., Sakurai, M.: Super-resolution system for 4k-hdtv. In: ICPR. pp. 4453–4458. IEEE (2014)
10. Gueguen, L., Sergeev, A., Kadlec, B., Liu, R., Yosinski, J.: Faster neural networks straight from jpeg. NeurIPS **31** (2018)
11. Haris, M., Shakhnarovich, G., Ukita, N.: Recurrent back-projection network for video super-resolution. In: CVPR. pp. 3897–3906 (2019)
12. Isobe, T., Jia, X., Gu, S., Li, S., Wang, S., Tian, Q.: Video super-resolution with recurrent structure-detail network. In: ECCV. pp. 645–660. Springer (2020)
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 (2018)
14. Kim, S.Y., Lim, J., Na, T., Kim, M.: 3DSRnet: Video super-resolution using 3d convolutional neural networks. arXiv preprint arXiv:1812.09079 (2018)
15. Kim, T.H., Sajjadi, M.S., Hirsch, M., Scholkopf, B.: Spatio-temporal transformer network for video restoration. In: ECCV. pp. 106–122 (2018)
16. Lai, W.S., Huang, J.B., Ahuja, N., Yang, M.H.: Deep laplacian pyramid networks for fast and accurate super-resolution. In: CVPR. pp. 624–632 (2017)
17. Li, S., He, F., Du, B., Zhang, L., Xu, Y., Tao, D.: Fast spatio-temporal residual network for video super-resolution. In: CVPR. pp. 10522–10531 (2019)
18. Li, W., Tao, X., Guo, T., Qi, L., Lu, J., Jia, J.: MuCAN: Multi-correspondence aggregation network for video super-resolution. In: ECCV. pp. 335–351. Springer (2020)
19. Li, X., Jin, X., Yu, T., Sun, S., Pang, Y., Zhang, Z., Chen, Z.: Learning omni-frequency region-adaptive representations for real image super-resolution. In: AAAI. vol. 35, pp. 1975–1983 (2021)
20. Li, Y., Jin, P., Yang, F., Liu, C., Yang, M.H., Milanfar, P.: COMISR: Compression-informed video super-resolution. In: ICCV (2021)

21. Liu, C., Yang, H., Fu, J., Qian, X.: Learning trajectory-aware transformer for video super-resolution. In: CVPR. pp. 5687–5696 (2022)
22. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV. pp. 10012–10022 (2021)
23. Lu, G., Ouyang, W., Xu, D., Zhang, X., Gao, Z., Sun, M.T.: Deep kalman filtering network for video compression artifact reduction. In: ECCV. pp. 568–584 (2018)
24. Lu, G., Zhang, X., Ouyang, W., Xu, D., Chen, L., Gao, Z.: Deep non-local kalman network for video compression artifact reduction. TIP **29**, 1725–1737 (2019)
25. Nah, S., Baik, S., Hong, S., Moon, G., Son, S., Timofte, R., Mu Lee, K.: Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In: CVPRW. pp. 0–0 (2019)
26. Qin, Z., Zhang, P., Wu, F., Li, X.: FcaNet: Frequency channel attention networks. In: ICCV. pp. 783–792 (2021)
27. Sajjadi, M.S., Vemulapalli, R., Brown, M.: Frame-recurrent video super-resolution. In: CVPR. pp. 6626–6634 (2018)
28. Tao, X., Gao, H., Liao, R., Wang, J., Jia, J.: Detail-revealing deep video super-resolution. In: ICCV. pp. 4472–4480 (2017)
29. Tian, Y., Zhang, Y., Fu, Y., Xu, C.: TDAN: Temporally-deformable alignment network for video super-resolution. In: CVPR. pp. 3360–3369 (2020)
30. Wang, X., Chan, K.C., Yu, K., Dong, C., Change Loy, C.: EDVR: Video restoration with enhanced deformable convolutional networks. In: CVPRW (2019)
31. Wang, Z., Liu, D., Chang, S., Ling, Q., Yang, Y., Huang, T.S.: D3: Deep dual-domain based fast restoration of jpeg-compressed images. In: CVPR. pp. 2764–2772 (2016)
32. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE TIP **13**(4), 600–612 (2004)
33. Xu, K., Qin, M., Sun, F., Wang, Y., Chen, Y.K., Ren, F.: Learning in the frequency domain. In: CVPR. pp. 1740–1749 (2020)
34. Xu, Y., Gao, L., Tian, K., Zhou, S., Sun, H.: Non-local convlstm for video compression artifact reduction. In: ICCV. pp. 7043–7052 (2019)
35. Xue, T., Chen, B., Wu, J., Wei, D., Freeman, W.T.: Video enhancement with task-oriented flow. IJCV **127**(8), 1106–1125 (2019)
36. Yang, F., Yang, H., Fu, J., Lu, H., Guo, B.: Learning texture transformer network for image super-resolution. In: CVPR. pp. 5791–5800 (2020)
37. Yi, P., Wang, Z., Jiang, K., Jiang, J., Lu, T., Tian, X., Ma, J.: Omniscient video super-resolution. In: ICCV. pp. 4429–4438 (2021)
38. Zeng, Y., Yang, H., Chao, H., Wang, J., Fu, J.: Improving visual quality of image synthesis by a token-based generator with transformers. NeurIPS **34**, 21125–21137 (2021)
39. Zhang, L., Zhang, H., Shen, H., Li, P.: A super-resolution reconstruction algorithm for surveillance images. Signal Processing **90**(3), 848–859 (2010)
40. Zheng, H., Fu, J., Mei, T., Luo, J.: Learning multi-attention convolutional neural network for fine-grained image recognition. In: ICCV. pp. 5209–5217 (2017)