


# Lightweight Video Super-Resolution for Compressed Video

Ilhwan Kwon, Jun Li and Mukesh Prasad \* 

School of Computer Science, FEIT, University of Technology Sydney, Sydney 2007, Australia

\* Correspondence: mukesh.prasad@uts.edu.au

**Abstract:** Video compression technology for Ultra-High Definition (UHD) and 8K UHD video has been established and is being widely adopted by major broadcasting companies and video content providers, allowing them to produce high-quality videos that meet the demands of today's consumers. However, high-resolution video content broadcasting is not an easy problem to be resolved in the near future due to limited resources in network bandwidth and data storage. An alternative solution to overcome the challenges of broadcasting high-resolution video content is to downsample UHD or 8K video at the transmission side using existing infrastructure, and then utilizing Video Super-Resolution (VSR) technology at the receiving end to recover the original quality of the video content. Current deep learning-based methods for Video Super-Resolution (VSR) fail to consider the fact that the delivered video to viewers goes through a compression and decompression process, which can introduce additional distortion and loss of information. Therefore, it is crucial to develop VSR methods that are specifically designed to work with the compression–decompression pipeline. In general, various information in the compressed video is not utilized enough to realize the VSR model. This research proposes a highly efficient VSR network making use of data from decompressed video such as frame type, Group of Pictures (GOP), macroblock type and motion vector. The proposed Convolutional Neural Network (CNN)-based lightweight VSR model is suitable for real-time video services. The performance of the model is extensively evaluated through a series of experiments, demonstrating its effectiveness and applicability in practical scenarios.

**Keywords:** video super-resolution; video compression; motion vector; spatio-temporal consistency



**Citation:** Kwon, I.; Li, J.; Prasad, M. Lightweight Video Super-Resolution for Compressed Video. *Electronics* **2023**, *12*, 660. <https://doi.org/10.3390/electronics12030660>

Academic Editor: Włodzimierz Kasprzak

Received: 22 December 2022

Revised: 19 January 2023

Accepted: 21 January 2023

Published: 28 January 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The last two decades have witnessed outstanding achievements in science and technology. Expensive devices and spectacular media contents which are enjoyed only by a minority are now available for almost every individual. Particularly, a camera device is nowadays common to everyone's daily life since it is embedded into portable devices such as smartphones, tablets, and laptops. Furthermore, the multimedia contents generated by the camera are digitalized, stored, shared, and transmitted rapidly and globally. Today the number of video content viewers watching over-the-top (OTT) media services such as Netflix, Roku, Disney+, YouTube, and Amazon Prime Video are rapidly increasing with the video streaming service's ongoing developments.

Meanwhile, multimedia content consumers prefer to watch higher resolution videos because of their enhanced vivid and realistic effects. The general video resolution on streaming services is FullHD (1920 × 1080) or 4K (3840 × 2160) and it has recently reached up to 8K resolution (7680 × 4320). The progress in video resolution is significant and sooner or later the current maximum resolution could also be substituted by a much higher resolution format. However, high-resolution video requires a large amount of storage, network bandwidth as shown in Table 1, and longer elapsed time to be transferred to clients through a network, inevitably resulting in a lower resolution video being delivered to viewers subject to network conditions. To overcome the problems of storage and bandwidth, streaming service providers have been developing various methods, such as

Per-Title Encoding, Adaptive Bitrate Streaming (ABR), Decentralized Content Delivery, Dynamic Optimizer, and Content Delivery Network (CDN).

**Table 1.** Recommended video bitrates for video stream service [1].

Type	Video Bitrate, Standard Frame Rate (24, 25, 30)	Video Bitrate, High Frame Rate (48, 50, 60)
2160p (4K)	35~45 Mbps	53~68 Mbps
1440p (2K)	16 Mbps	24 Mbps
1080p	8 Mbps	12 Mbps
720p	5 Mbps	7.5 Mbps
480p	2.5 Mbps	4 Mbps
360p	1 Mbps	1.5 Mbps

In addition, the abundant amount of pre-existing low-resolution video content also needs to be delivered to OTT service consumers. However, the video quality of the old low resolution is not satisfiable to be played on Ultra-High Definition (UHD) size display devices at home. Accordingly, if the received low-resolution video on higher resolution display devices including smartphones can be converted to high-resolution video clips delicately, consumers could enjoy the benefits of high-speed video streaming as well as better quality videos than before.

The aforementioned network bandwidth, storage, and video quality issues can be resolved with Video Super-Resolution (VSR) technology which reconstructs high-resolution video from a lower resolution video through the use of various features in one or sequential frames in the video [2]. It starts from conventional computer vision technology including static interpolation theories such as the nearest neighbor, bilinear, or bicubic filter, and has shown a significant progress by adopting Convolutional Neural Network (CNN). CNN is a type of deep learning neural network that is particularly well suited for image processing tasks. The key idea behind CNN is to use convolutional layers, which scan the input image with a small filter (also called kernel or weights) and applies the same transformation at each location of the image. By applying multiple convolutional layers, a CNN can learn increasingly complex features of the image, such as edges, textures, and patterns. Many researchers have demonstrated that CNN-based Super-Resolution methods produce clearer and higher resolution output when compared to traditional interpolation techniques [3–5]. The latest research on VSR demonstrated meaningful advances in terms of the quality of super-resolved video and conversion speed.

The four principal streams of the related research are: (1) Recurrent Frame-based VSR Network (FRVSR, RBPN, RRN) [6–8], (2) Spatio-Temporal VSR Network (SOF-VSR, STVSR, TDAN, TOFlow, TDVSR-L) [9–13], (3) Generative Adversarial Network (GAN)-based SR Network [14–17], and (4) Video Compression-informed VSR Network (FAST, COMISR, CDVSR, CIAF) [18–21].

This paper proposes a method utilizing the information from the compressed video to achieve a lightweight VSR model applicable in video streaming services without seriously decreasing the quality of super-resolved video, namely Compression-informed Lightweight VSR (CILVSR), as shown briefly in Figure 1. This study aims to utilize more information acquired from the procedure of video decoding to improve the performance of the VSR model in terms of speed and lightness. The information covers slice type, which is almost similar to frame type in here, macroblock type, group of pictures, and motion vector.

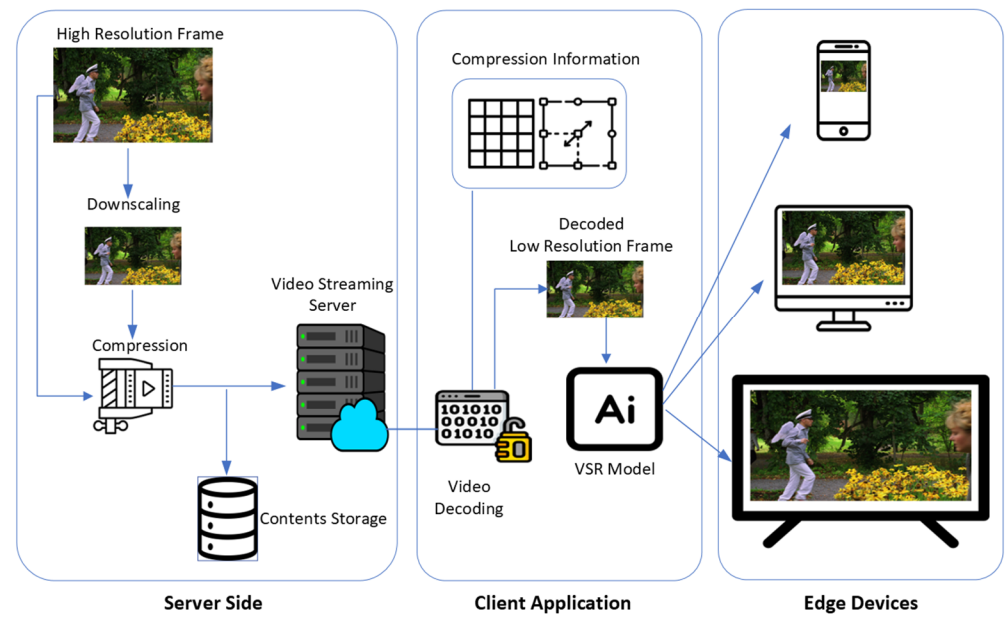


Figure 1. Overview of proposed method in video streaming service.

Table 2 presents slice type of the H.264 Video codec standard [22]. The slice type of Intra-frame is either I (Intra) slice or SI (Switching I) slice and the slice type of Inter-frame is P (Predictive), SP (Switching P), B (Bi-directional Predictive), or SB (Switching B) type.

Table 2. H.264 slice type.

slice_Type	Name	slice_Type	Name
0	P (P slice)	5	P (P slice)
1	B (B slice)	6	B (B slice)
2	I (I slice)	7	I (I slice)
3	SP (SP slice)	8	SP (SP slice)
4	SI (SI slice)	9	SI (SI slice)

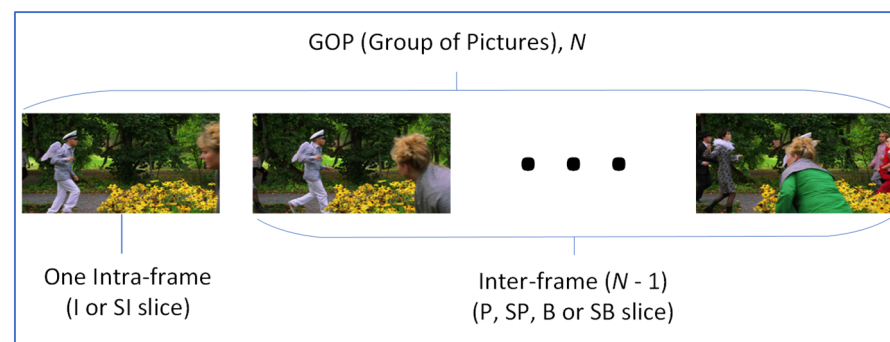
The iterative period of selecting the Intra-frame group of pictures(GOP), i.e., the gap between Intra-frames shown in Figure 2, is determined by encoding configuration. The small number of GOP means the target bitrate of the encoding file is big enough and this configuration is chosen in case a frequent scene change occurs in a video or a video content requires more details and less block noise after decoding. In the case of H.264, the start frame is designated as an Instantaneous Decoder Refresh (IDR) frame with Intra-frame type and once a frame is encoded with this frame type, all the statuses such as reference picture buffer status, reference frame number, and picture order count are initialized in decoding [22].

The proposed VSR model is composed of two main networks which are the Intra-frame-based network and Inter-frame-based network. The Intra-frame-based network utilizes periodic Intra-frames in compressed video and it is trained without any dependency on consecutive frames. As this network model adopts Intra-frames which consist of a significant amount of information in compressed video such as one still image, it is possible to be considered as a single image SR network which is beneficial to implement a lightweight VSR model. Meanwhile, the Inter-frame-based network presented in this study facilitates two consecutive frames for training to utilize the temporal relation between the Inter-frames. In this Inter-frame-based network training, the motion compensation process exploits the motion vector, macroblock type, and the completely decoded previous frame as a reference frame. Furthermore, the integration of the two models enables it to

be a simple and adaptable model, which utilizes the intact information from the original compressed video.

The contributions of this research are as follows:

- The VSR model in this paper consumes low computational resources for inference work without significantly damaging the quality of the video through adopting a smaller number of reference frames compared to other Spatio-Temporal-based VSR models;
- To extend the availability of VSR model under a bad network environment, the proposed model is separable by frame type;
- The proposed VSR model is appropriate for real-time video streaming services by using various information from a video decoder.



**Figure 2.** GOP in compressed video.

## 2. Related Works

This describes four types of VSR and analyzes their benefits and enhanced aspects of them [23].

### 2.1. Recurrent Frame-Based VSR Network

The Recurrent Frame-based VSR [24] Network improves upon traditional VSR methods by integrating single image super-resolution (SISR) and multi-image super-resolution (MISR) into a unified framework. SISR and MISR are able to extract missing information or details from other frames. SISR extracts different feature maps representing the target frame, while MISR offers multiple sets of feature maps from various frames. Deep SISR was proposed [3], which required a predefined upsampling operator. Improving upon this, methods such as progressive upsampling [4], residual learning [5,25,26], back-projection [27], upsampling layers [28], and recursive layers [29,30] have been introduced. In general, VSR using multiple frames focuses on two aspects: aligning frames with large motions and effectively fusing different frames. The Detail Fusion (DF) Network in [31] effectively fuses image details after aligning them using sub-pixel motion compensation (SPMC).

### 2.2. Spatio-Temporal VSR Network

Spatio-Temporal VSR Networks utilize both spatial and temporal features in a low-resolution video to recover high-resolution video by using multiple low-resolution frames as input data [32]. These include Super-resolve Optical Flows for Video SR (SOF-VSR) [9], Space-Time Video Super-Resolution (STVSR) [10], Temporally Deformable Alignment Network (TDAN) [11], Task-Oriented Flow (TOFlow) [12], Deep Dual Attention Network (DDAN) [33], and Temporal Dynamics VSR (TDVSR-L) [13]. The principal methodology of VSR networks is frame alignment between the target frame and reference frame at first by adopting dynamic kernels in convolution. After this process, based on the aligned frames and the reference frame, the networks realize a better quality of a high-resolution frame with convolution and reconstruction layers.

In the case of STVSR, it is capable of predicting a high frame rate (HFR) and HR frames without explicit interpolation of intermediate low-resolution frames. Furthermore, by training temporal interpolation as well as spatial super-resolution simultaneously, STVSR



can synthesize an HR slow-motion video from lower resolution video. STVSR is composed of three networks, a feature temporal interpolation network, a deformable ConvLSTM (Convolutional Long Short-Term Memory), and a deep reconstruction network. The benefit of STVSR is that it proposes one-stage interpolation to synthesize low-resolution feature maps for missing frames without requiring explicit supervision.

### 2.3. GAN Based SR

Generative Adversarial Network (GAN) is a type of neural network architecture consisting of two parts: a generator and a discriminator. The generator creates new, synthetic data, while the discriminator attempts to distinguish the synthetic data from real data. The two parts are trained together, with the generator trying to create data that can fool the discriminator, and the discriminator trying to correctly identify the synthetic data. GAN is widely employed in various applications within the fields of image classification and image conversion [34–36]. Another latest research using GAN is to improve the performance of image and video super-resolution with a high-frequency discriminator to reconstruct low-resolution images to their related high-resolution images. Although, the performance evaluation of super-resolution reconstructing has been decided with PSNR and SSIM, [14] indicating that PSNR could not be consistent with the results of a human eye. Hence, NIQE [37] and Ma [38] are broadly used for super-resolution problems. However, it was noted that those perception-based indicators were not appropriate to evaluate image recovery [39–43]. Therefore, the Learned Perceptual Image Patch Similarity (LPIPS) method [44] is suggested whilst it compensates for the perceptual limit of other indicators by comparing the feature similarity between super-resolved image and the ground truth image.

GAN was first utilized by [45] proposing EnhancedNET for super resolution. Additionally, [46] introduced perceptual loss for the developed realism of the image. However, a substantial amount of noise was produced when attempting to improve the authenticity of the image. Thus, while [14] redesigned both the generator and the differentiator, ES-RGAN [15] changed the generator in Super-Resolution Generative Adversarial Network (SRGAN) to Residual-in-Residual Dense Block (RRDB) [47]. Furthermore, recent RankSRGAN [16] has used external data to form ranker constraints based on reconstructing images by optimizing its own reproduced image. These GAN-based models have utilized either perceptual loss [46] or combined Ranker for generating and constraining the high-frequency details of the reconstructed frame. These constraints tended to be indirectly affected by the featured layer of the image rather than directly by the high-frequency information of the image. Due to this reason, it was concluded that the constraints showed less efficiency.

The work related to dual discriminator was suggested by [48] for video generation and was known as one of the most effective video generation frameworks. The author of [17] shows another approach based on GAN. This is motivated by recurrent back-projection networks (RBPNs) using back-projection for multi-image super-resolution whilst it manipulates temporal information in neighbor frames in a video to acquire better performance. This research especially exploits several losses to achieve optimal perceptual image quality as well as PSNR improvements such as mean square error (MSE) loss, perceptual loss, adversarial loss, and total variation loss.

### 2.4. Video Compression Informed VSR

To combine the technology of video compression with super-resolution methods, a few approaches are proposed recently such as Free Adaptive Super-resolution via Transfer (FAST) [18], Compression-informed Video Super-Resolution (COMISR) [19], Compressed Domain Video SR (CDVSR) [20], and Codec Information Assisted Framework (CIAF) [21]. The FAST framework suggests that an initial SR frame from a compressed video is transferred to consecutive frames; therefore, exploiting the temporal correlation between adjacent frames and extracting motion vectors with residuals in the compressed video enable the upsampling of sequential frames fast and efficiently. The idea can also be enhanced

by considering SR for non-overlapping blocking and removing artifacts by an adaptive deblocking filter in the SR procedure [49]. The experiment using FAST with HEVC compressed 20 video sequences was 15 times faster than SRCNN [3], KRR [50], and ANR [51], meaning that real-time SR for UHD video contents could be embedded soon in various devices such as TV, smartphones, and tablets. Furthermore, if FAST overcomes the limit that requires a periodic frame reset for decent results and abrupt scene change in a video, the architecture integration of video compression and FAST will bring significant synergy in the video streaming industry.

Another model using video compression information is COMISR [19] which consists of a flow estimation module, bi-directional recurrent module, and Laplacian enhancement component. It shows good performance of SR outcomes under various compression levels. The bi-directional recurrent module in COMISR produces a predicted HR frame and the detailed flow estimation module assists in conserving high-frequency details of HR flow. Additionally, the Laplacian enhancement module is added to restore the fine details of compressed video. The evaluation result shows that recovering more details such as the texture of low-resolution pictures as well as generating denoised video having fewer artifacts is possible by this model.

CDVSR [20] is the VSR model utilizing intrinsic information from the decoded video to acquire a super-resolved frame. This model is composed of a Guided Coding Prior Injection (GCPI) feature extraction module, Motion Vector Guided Soft Alignment (MV-GSA) module, Attention-Aided Temporal Fusion scheme, and Cross-Scale SR Reconstruction network. This model proposes an efficient strategy for acquiring better performance by using prior decoding and a feature alignment scheme. It is one of the decent models showing the synergy effect of the integration of video compression technology into video super-resolution.

CIAF [21] suggests a method to enhance the recurrent based VSR models for compressed videos. Information is reused such as motion vector and residual from decoded video to improve the efficiency of the recurrent based VSR model. Specifically, this framework proposes a motion vector alignment technique to acquire better SR frame quality on pre-existing recurrent VSR models.

These methods that use video compression information demonstrate good performance in terms of efficiency and expandability compared to other types of VSR models in 2.1 to 2.3. However, several barriers utilizing these models in a real-time environment still exist such as low-end devices with outdated GPU or a small size of memory inside because they are not composed of simple architecture or low weight parameters which are necessary for real-time inferencing.

To solve such issues as mentioned above, this paper suggests lightweight VSR model architecture using simple layers as well as video codec information to increase the computational resource efficiency in the process of inference.

### 3. Methodology

The proposed method, CILVSR, consists of two main parts to obtain super-resolved frames from a low-resolution video, Intra-frame upsampling and Inter-frame upsampling.

Firstly, after decoding an encoded LR video, GOP and frame type information are acquired. Based on the information, frames are classified as Intra-frame or Inter-frame ahead. The super-resolved results of all frames in GOP,  $SR_{GOP}$  in Equation (1) are a group of the estimated high resolution of Intra-frame,  $SR_{intra}$  and a group of the estimated high resolution of Inter-frames,  $SR_{inter}^k$ :

$$SR_{GOP} = SR_{intra} + \sum_{k=1}^{N-1} SR_{inter}^k \quad (1)$$

where  $N$  is the number of frames in a GOP.

The upsampling of low-resolution Intra-frame,  $I_{intra}^{LR}$  can be considered a single image super resolution (SISR) method. In general, SISR training and inference process require fewer computing resource than a VSR model using multiple frames for training and

inference. Extracting the Intra-frame from the encoded video and treating it as a separate SISR module is profitable to reduce the overall burden of VSR model training. In this paper, SRCNN [6] with Laplacian enhancement is used to obtain Super-resolved Intra-frames,  $I_{intra}^{SR}$  from compressed low-resolution video. Laplacian enhancement is adopted to restore high-frequency details which are reduced in the process of video encoding [36]. A Laplacian enhanced frame is produced by a Gaussian kernel blur  $G(\cdot, \cdot)$ :

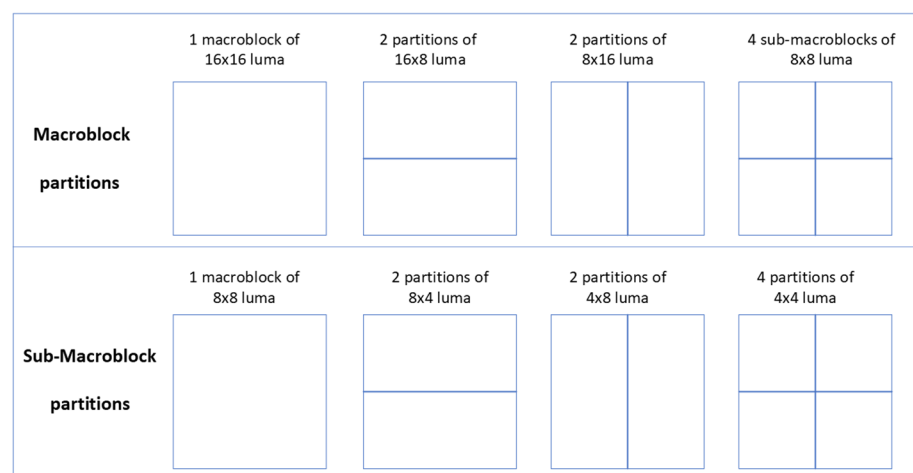
$$I_{intra}^{SR} = I_i^{HR} + \alpha \left( I_i^{HR} - G \left( I_i^{HR}, \sigma = 1.5 \right) \right) \tag{2}$$

where  $\sigma$  is the width of the Gaussian kernel and  $I_i^{HR}$  is an intermediate HR frame.

Meanwhile, in video compression, the relation between Inter-frames is described as predicted frames (P frames) or bi-directional predicted frames (B frames). P frames refer to previous frames for compression/decompression and it contains two types of macroblock, intra macroblock (I\_\* in Table 3) and predicted macroblock (P\_\* in Table 3). Table 3 shows the macroblock type for the P frame in H.264. A macroblock is a unit of pixels for video compression and its basic size in the case of H.264 video compression standard is  $16 \times 16$  pixels. Each macroblock can be partitioned into a sub-macroblock such as  $16 \times 8$ ,  $8 \times 16$ ,  $8 \times 8$  or  $4 \times 4$  to acquire better compression results as Figure 3.

**Table 3.** Macroblock types for P frame.

mb_Type	Name of mb_Type
0	P_L0_16 × 16
1	P_L0_L0_16 × 8
2	P_L0_L0_8 × 8
3	P_8 × 8
4	P_8 × 8ref0
inferred	P_Skip
5	I_4 × 4
6~29	I_16 × 16
30	I_PCM



**Figure 3.** Macroblock partitions in H.264.

The first step of Inter-frame encoding is to find motion vectors of similar pixel blocks between frames and the second step is the bitwise compression of the residuals and the estimated motion vectors which are encoded with the value of mb\_type from 0 to

4 in Table 3. Throughout this Inter-frame compression process, temporal redundancy between frames in GOP is eliminated because in case the same macroblocks exist between consecutive frames, the macroblocks can be compressed and represented after decoding as a type of reference macroblock and motion vectors instead of encoding all the macroblocks in sequential frames.

In the case of JM (Joint Model) software [52] for H.264 encoding, to find out motion vector, Unsymmetrical Multi-Hexagon Search (UMHexagonS), and Center Biased Fractional Pel Search (CBFPS) methods are used for high-speed motion estimation. This search process is the most time-consuming module in video encoding because it requires numerous mathematical calculations. To achieve a high compression ratio of Inter-frames, finding the exact motion vectors is crucial in the video encoder.

In the meantime, optical flow is also a vintage technique for finding out moving patterns in two frames occurring by object movement or lighting change [53]. To implement the high performance of the VSR model, many researchers adopt the optical flow technique to realize the Spatio-Temporal correlation between frames and a substantial amount of evaluation results from the research show the effectiveness of using optical flow for VSR. Inspired by the SOF-VSR model [9] and VESPCN [54] utilizing an optical flow network for video upscaling, the proposed method utilizes the motion vector of compressed low-resolution video as one form of input data for the model training of Inter-frames.

Figure 4 shows the overall architecture proposed. The architecture is mainly composed of two networks, SRCNN with Laplacian enhancement for Intra-frame upsampling in Table 4 [3] and Spatio-Temporal ESPCN [54] with Laplacian Enhancement for Inter-frame upsampling in Table 5. In Figure 4,  $I_r^{LR}$  and  $I_c^{LR}$  are decoded low-resolution frames.  $I_r^{LR}$  means a low-resolution reference frame which is a frame that is used as a reference frame to attain motion vector and  $I_c^{LR}$  is a low-resolution current frame that should be super-resolved.  $I_c^{LR}$  is classified into Intra-frame or Inter-frame by frame type information from a video decoder. Other necessary information for the proposed model such as the number of GOP, macroblock type, motion vector, and reference frame number in each macroblock can also be extracted from a video decoder. In general, the information is already transferred to the display module before the initiation of a video streaming playback. In other words, there is no additional burden to acquire the information for VSR model training or inference process.

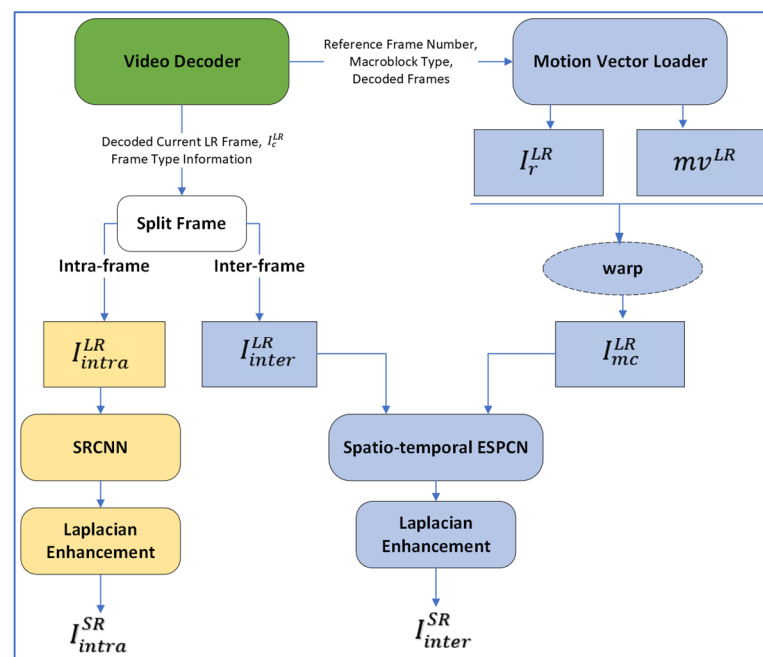


Figure 4. Overview of CILVSR network.

**Table 4.** SRCNN architecture.

Layer	SRCNN
1	Conv2d (1, 64, kernel_size = 9, padding = 4)/ReLU
2	Conv2d (64, 32, kernel_size = 5, padding = 2)/ReLU
3	Conv2d (32, 1, kernel_size = 5, padding = 2)/ReLU

**Table 5.** Spatio-Temporal ESPCN (9 Layer Early Fusion Network).  $N_c$  means the number of input image channels, 1 in this model and  $N_s$  is the number of sequential frames used for network inputs, 2 in this model.

Layer	Spatio-Temporal ESPCN
1	Conv2d ( $N_c \times N_s$ , 64, kernel_size = 3, padding = 1)/LeakyReLU
2	Conv2d (64, 64, kernel_size = 3, padding = 1)/LeakyReLU
3	Conv2d (64, 64, kernel_size = 3, padding = 1)/LeakyReLU
4	Conv2d (64, 32, kernel_size = 3, padding = 1)/LeakyReLU
5	Conv2d (32, 32, kernel_size = 3, padding = 1)/LeakyReLU
6	Conv2d (32, 32, kernel_size = 3, padding = 1)/LeakyReLU
7	Conv2d (32, 20, kernel_size = 3, padding = 1)/LeakyReLU
8	Conv2d (20, $N_c \times \text{Scale}^2$ , kernel_size = 3, padding = 1)/LeakyReLU/ PixelShuffle(Scale)
9	Conv2d ( $N_c$ , $N_c$ , kernel_size = 1, padding = 0)/LeakyReLU

General video compression standards support multiple reference frames and bi-directional referencing to obtain one motion vector, but the proposed method considers only single previous frame referencing in this research to implement the most lightweight VSR model.

The motion vector loader in Figure 4 fetches the motion vector elements from decompressed low-resolution video following the macroblock types. It is unlikely that optical flow representing every pixel in a frame, one motion vector pair,  $mv^p(mv_x, mv_y)$ , can represent from a  $4 \times 4$  pixel partition to a  $16 \times 16$  pixel partition in the case of H.264 [22], as shown in Figure 3. For example, if a macroblock of  $16 \times 16$  pixels is partitioned into four  $8 \times 8$  pixel units, four motion vector pairs can be allocated to express the movement of one macroblock such as the upper right partition in Figure 3. Furthermore, if the sub-macroblock partition type is an  $8 \times 8$  pixel block and the block can be partitioned to a  $4 \times 4$  macroblock type, then the total number of motion vector pairs in a macroblock is 16.

$$MV^{LR} = MVL \left( \sum_{p=1}^M mv^p; mb_{type} \right) p = 1 \sim M, \quad (3)$$

where  $MVL$  is a motion vector loader,  $MV^{LR}$  is motion vector values from low-resolution video and  $mb_{type}$  is the macroblock type of each macroblock. The maximum number of motion vector pairs,  $M$  is 16 as aforementioned in the case of H.264.

The subsequent process is motion compensation using a reference frame and folded low-resolution motion vector. Similar to the motion compensation module in [54], motion-compensated frames are acquired through warping with LR grids and the extracted motion vectors from the decoder.

$$I_{mc}^{LR} = W \left( I_r^{LR}, MV^{LR} \right) \quad (4)$$

where  $W$  is a warping module that utilizes bilinear interpolation-based grid-sampling with inputs as a reference frame,  $I_r^{LR}$  and motion vector, and  $MV^{LR}$  of the low-resolution frame.

The output frame,  $I_{mc}^{LR}$  from the motion compensation block and current low-resolution frame,  $I_{inter}^{LR}$  are fed into the multi-frame-based ESPCN in [54]. As Table 5 shows specifically,



the 9L-E3-MC network in [54] was adopted and it consists of eight  $3 \times 3$  convolutional layers, a pixel shuffle layer and one  $1 \times 1$  convolutional layer.

$$I_{inter}^{SR} = Net_{multi-frame-espcn}(I_{inter}^{LR}, I_{mc}^{LR}; \theta_{SR}), \quad (5)$$

where  $\theta_{SR}$  is the set of parameters of Spatio-Temporal ESPCN layers.

To train Inter-frames with multi-frame-based ESPCN, MSE loss with Laplacian enhancement and Huber loss are adopted similarly with [54]. The Huber loss constrains motion vector values while training as it works to treat optical flows.

$$\mathcal{L}_{inter}(\theta^*, \theta_{\Delta}^*) = \underset{\theta, \theta_{\Delta}}{\operatorname{argmin}} \left\| I_{inter}^{HR} - I_{inter}^{SR} \right\|_2^2 + \beta \left\| I_{mc}^{LR} - I_{inter}^{SR} \right\|_2^2 + \lambda \mathcal{H}(\theta_x, y, \Delta_c) \quad (6)$$

where  $\theta_{\Delta}$  is model parameters,  $\beta$  is the coefficient for the motion compensation module,  $\lambda$  is the coefficient for Huber loss and  $I_{inter}^{HR}$  is the ground truth of Inter-frame. Similarly, to achieve a lightweight VSR model, different from SOF-VSR [9] or VESPCN [54], the proposed model utilizes motion vectors of two decoded frames directly instead of utilizing optical flows from three frames.

Meanwhile, for the training of the SRCNN model with Laplacian enhancement, MSE loss is used and the learning rate of the first two layers is  $10^{-4}$ . This value is 10 times bigger than the last layer following the suggestion from the original SRCNN model [3].

$$\mathcal{L}_{intra}(\theta^*, \theta_{\Delta}^*) = \underset{\theta, \theta_{\Delta}}{\operatorname{argmin}} \left\| I_{intra}^{HR} - I_{intra}^{SR} \right\|_2^2 \quad (7)$$

## 4. Experiments

### 4.1. Training and Evaluation Dataset

The Consumer Digital Video Library (CDVL) dataset [55] is used for training such as SOF-VSR [9] and VESPCN [54]. Furthermore, following [9], 145 full HD videos is downsampled to the size of  $960 \times 540$  with MATLAB bicubic downsampling library denoted as BI in [9]. Additionally, the downsampled videos are converted to a sequence of uncompressed png files. To acquire a similar environment of video streaming, (1) the png files are encoded to h.264 videos with JM encoder, and (2) decoded the h.264 videos and acquired a sequence of decoded png files. The configuration of the encoder for the training dataset is (1) an h.264 baseline profile, (2) the GOP period is 15, (3) the QP for I and P frame is 28, (4) the Max search range for motion estimation is 32, (5) the number of previous frames used for the inter motion search is 1, and (6) rate control is disabled to acquire high bitrate encoded files.

This paper does not consider various dataset cases which select several CRF values to verify the relation between encoded bitrate and output quality because the correlation between them is already verified in [19,20,56].

The quality of the decoded png files is slightly degraded compared to the png files for video encoding. For testing the result of the trained model, the Vid4 dataset is used because the benchmark dataset including over 34 sequential frames is widely utilized in many papers to compare the performance of various VSR models. As with training data, to realize a similar environment with video streaming, the Vid4 benchmark dataset is also encoded with an h.264 JM encoder and decoded to obtain uncompressed png files. The configuration of the encoder for the test dataset is the same as the configuration for generating the training dataset.

### 4.2. Experiment Environment

The code for the experiment is implemented in PyTorch 1.9.0 framework and the training job is performed with an RTX 6000, 24GB. Similar to [2], the Adam optimizer with  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$  is selected and because of an issue regarding decompressed frame indexing, the batch size is 1 for training. Increasing the batch size is not efficient for the

proposed networks because two types of video frames, Intra-frame and Inter-frame, are trained within a GOP and if the batch size is larger than GOP, a lot of frames should be skipped to avoid discontinuity in training data. The initial learning rate is set to  $1 \times 10^{-4}$  and divided by 2 after every 60 K iterations. The iteration number of 1 epoch is 200 K and the maximum number of the epoch is 100.

### 4.3. Evaluation on the Vid4 Dataset

#### 4.3.1. Quantitative Evaluation

The test result with the Vid4 dataset of the proposed model in comparison with the SOF-VSR model, which is one of the optical flow-based high-performance models, is shown in Tables 6–8. The test dataset of consecutive frames in each video is selected for evaluation. The PSNR value from the pre-trained weight and official code of SOF-VSR is 25.97 dB for the  $\times 4$  scale is insignificantly smaller than the paper [9] and it is suggested that the Vid4 dataset is used after decoding. The SSIM value from SOF-VSR is 0.77 and both PSNR and SSIM are superior to the proposed model. However, in the case of parameter number and FLOPs, the proposed model is much smaller and lighter for inference whilst it shows that the proposed model is more suitable than SOF-VSR in a real-time video streaming environment. Moreover, for the inference speed test with the Vid4 dataset, the SOF-VSR model consumes a lot of time for inferencing one frame on a moderate GPU machine such as NVidia 1050 Ti. On average, it requires over 200 msec per frame of Vid4 dataset inference for  $\times 4$  scale upsampling whilst the proposed model consumes approximately 40 msec for the same operation. This indicates that the proposed model is more appropriate in practical usage scenarios of the VSR model under a real-time environment than SOF-VSR. If the inference time of the VSR model is bigger than 60 msec, i.e., less than 15 frame per second (fps), the model is practically hard to be adopted as a solution in the consumer market.

**Table 6.** Comparative result of PSNR value of proposed method and other technique. The results marked with an asterisk (\*) are sourced from the respective papers.

Scale	Vid4 Dataset	SOF-VSR [9]	VESPCN [54]	Bicubic [57]	CILVSR
$\times 4$	Calendar	22.72	-	20.26	20.88
	City	26.73	-	25.52	24.85
	Foliage	25.52	-	23.00	23.46
	Walk	29.92	-	25.40	26.36
	Average	25.97	25.35 *	23.54	23.89

**Table 7.** Comparative result of SSIM value of proposed method and other technique. The results marked with an asterisk (\*) are sourced from the respective papers.

Scale	Vid4 Dataset	SOF-VSR [9]	VESPCN [54]	Bicubic [57]	CILVSR
$\times 4$	Calendar	0.74	-	0.59	0.60
	City	0.74	-	0.54	0.58
	Foliage	0.72	-	0.51	0.56
	Walk	0.88	-	0.76	0.79
	Average	0.77	0.76 *	0.60	0.63

**Table 8.** Comparative result of parameter number, FLOPs, and average inference time of proposed method and other technique. The results marked with an asterisk (\*) are sourced from the respective papers.

Scale		SOF-VSR [9]	VESPCN [54]	CILVSR
×4	Parameter number	1M	-	121K
	FLOPs	112.5G	14.0G *	2.8G
	Average inference time per Frame (msec)	240	-	40
×3	Parameter number	1.1M	-	119K
	FLOPs	205.0G	24.23G *	5.0G
	Average inference time per Frame (msec)	337	-	50

Meanwhile, the proposed model was also compared to the pre-trained VESPCN model, as shown in Tables 6–8. The proposed model demonstrates an advantage in terms of FLOPs consumption, which directly affects the inference speed. As a result, the proposed model has a faster inference time compared to the VESPCN model. Additionally, the proposed model only requires two consecutive frames for inference, while VESPCN needs three frames. This makes the proposed model more efficient in terms of frame utilization. This is particularly important in real-time video streaming scenarios where the number of frames required for inference can impact the overall performance due to sequential frame loss under unstable network conditions. Additionally, when comparing the performance of the proposed model with the VESPCN model, it is important to consider several factors. One such factor is that the VESPCN model was trained on the original full HD (1920 × 1080) dataset, whereas the proposed model was trained on a qHD (960 × 540) downsampled CDVL dataset. Additionally, the test results reported in the VEPCN paper were obtained using the original Vid4 dataset before decoding, which means that the input frames used to evaluate the VEPCN model were of higher quality than those used to evaluate the proposed model. These factors should be taken into account when interpreting the performance differences between the two models.

#### 4.3.2. Qualitative Evaluation

Figures 5 and 6 illustrate a visual comparison of traditional upscaling methods and the proposed CILVSR model. The blue box in the original high-resolution frame is highlighted in the zoomed-in versions below, which were obtained by applying various up-sampling methods. The proposed model outperforms traditional interpolation methods. Figures 7 and 8 demonstrate a visual comparison of the SOF-VSR model with the proposed CILVSR model. While the output visual quality of the CILVSR model may not be as high as that of the SOF-VSR model when utilizing ×2 upscaling, it is still able to produce almost similar quality. To ensure fairness, the CILVSR model was trained using the same dataset as the SOF-VSR model, i.e., 6392 png files, bicubic downsampled (BI) from 145 CDVL video clips. As a further study, if more data are added for model training or low-resolution video after processing anti-aliasing, a better quality of frame from the proposed model will possibly be acquired without harming the advantage of this model as a lightweight VSR model.



Figure 5. Visual comparison of 3× upsampling results on image sample “Calendar”, ground truth (GT), bilinear, nearest neighbor, Lanczos [58], bicubic [57], and CILVSR.



Figure 6. Cont.





**Figure 6.** Visual comparison of  $3\times$  upsampling results on image sample “Walk”, ground truth (GT), bilinear, nearest neighbor, Lanczos [58], bicubic [57], and CILVSR.



**Figure 7.** Visual comparison of  $2\times$  SR results on image sample “City”, SOF-VSR (Left) and CILVSR (Right).



**Figure 8.** Visual comparison of  $2\times$  SR results on image sample “Foliage”, SOF-VSR (Left) and CILVSR (Right).



#### 4.4. Result Analysis

The deep learning-based Video Super-Resolution (VSR) model can be challenging to apply in a client application in real-time environments due to the limited resources such as computational power, bandwidth, and storage. Therefore, it is essential to consider two main requirements for client-side implementation of VSR: inference speed and integration efficiency. Firstly, the video client application should meet necessary requirements such as a minimum fps of 15 fps, and a small size of weight file for deployment. Existing VSR models fall short in these areas as they primarily focus on output image quality. In addition, in a streaming service or video player, the VSR module's low-resolution input frames are provided by the video codec decoder module. This means that the sequential frame feeding of the VSR module relies on the video codec module in a real environment. Thus, using fewer frames for the VSR model is beneficial to avoid complexity and errors that occur during the integration process. While other lightweight-related papers focus on hardware-based lightweight solutions [59] or using codec information as a calibration factor [60], this approach focuses on elements in the VSR model architecture. Additionally, the proposed model has a separable architecture as Intra-frame-based network and Inter-frame-based network according to circumstances. This type of network structure will be helpful in the worst network environment such as an intermittent network disconnection situation by selecting only an Intra-frame-based network and displaying restored SR frames.

#### 5. Conclusions

The current study demonstrated a lightweight VSR model and related architectures. It showed that most previous deep learning-based VSR focused on the performance of Super-Resolved Video in terms of PSNR and SSIM which were acquired by complex CNN architecture with recurrent blocks, residual module connection, flow estimation, and so forth. The models require a significant amount of computational resources for training and the output model contains a large number of parameters inside. These complexities obstruct the expansion of VSR in the real world as a state-of-the-art solution to solve the issues of network and storage deficiency. Thus, the direction of the latest research regarding VSR is developing much faster to produce more practical outcomes through the use of various spatial and temporal information from the compressed video. The trend of this research further reveals that neither PSNR nor SSIM could be the only index to measure the performance of VSR. In other words, other factors such as the time to train the VSR model, the agility of the pre-trained model, inference speed, and perceptual loss should be considered to determine the performance of the VSR model. Throughout this research, the methodology to implement a light, practical, and high-performance architecture in terms of the above factors along with the further progression of performance is proposed.

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

**Funding:** This research received no external funding.

**Data Availability Statement:** <https://www.cdv1.org/> (accessed on 21 December 2022), <https://github.com/YounggjuuChoi/Deep-Video-Super-Resolution/blob/master/Doc/Dataset.md> (accessed on 21 December 2022).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Ant Media. Available online: <https://antmedia.io/video-bitrate-vs-resolution-4-key-differences-and-their-role-in-video-streaming/> (accessed on 12 October 2022).
2. Liborio, J.D.; Melo, C.; Silva, M. Internet Video Delivery Improved by Super-Resolution with GAN. *Future Internet* **2022**, *14*, 364. [CrossRef]
3. Dong, C.; Loy, C.C.; He, K.; Tang, X. Image Super-Resolution Using Deep Convolutional Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *38*, 295–307. [CrossRef]
4. Lai, W.; Huang, J.; Ahuja, N.; Yang, M. Deep Laplacian Pyramid Networks for Fast and Accurate Super-Resolution. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5835–5843.
5. Kim, J.; Lee, J.K.; Lee, K.M. Accurate Image Super-Resolution Using Very Deep Convolutional Networks. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2015; pp. 1646–1654.
6. Sajjadi, M.S.; Vemulapalli, R.; Brown, M.A. Frame-Recurrent Video Super-Resolution. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6626–6634.
7. Haris, M.; Shakhnarovich, G.; Ukita, N. Recurrent Back-Projection Network for Video Super-Resolution. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 3892–3901.
8. Isobe, T.; Zhu, F.; Jia, X.; Wang, S. Revisiting Temporal Modeling for Video Super-resolution. *ArXiv* **2020**, arXiv:abs/2008.05765.
9. Wang, L.; Guo, Y.; Liu, L.; Lin, Z.; Deng, X.; An, W. Deep Video Super-Resolution Using HR Optical Flow Estimation. *IEEE Trans. Image Process.* **2020**, *29*, 4323–4336. [CrossRef]
10. Xiang, X.; Tian, Y.; Zhang, Y.; Fu, Y.R.; Allebach, J.P.; Xu, C. Zooming Slow-Mo: Fast and Accurate One-Stage Space-Time Video Super-Resolution. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 3367–3376.
11. Tian, Y.; Zhang, Y.; Fu, Y.R.; Xu, C. TDAN: Temporally-Deformable Alignment Network for Video Super-Resolution. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2018; pp. 3357–3366.
12. Xue, T.; Chen, B.; Wu, J.; Wei, D.; Freeman, W.T. Video Enhancement with Task-Oriented Flow. *Int. J. Comput. Vis.* **2017**, *127*, 1106–1125. [CrossRef]
13. Liu, D.; Wang, Z.; Fan, Y.; Liu, X.; Wang, Z.; Chang, S.; Wang, X.; Huang, T.S. Learning Temporal Dynamics for Video Super-Resolution: A Deep Learning Approach. *IEEE Trans. Image Process.* **2018**, *27*, 3432–3445. [CrossRef]
14. Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Aitken, A.P.; Tejani, A.; Totz, J.; Wang, Z.; Shi, W. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2016; pp. 105–114.
15. Wang, X.; Yu, K.; Wu, S.; Gu, J.; Liu, Y.; Dong, C.; Loy, C.C.; Qiao, Y.; Tang, X. ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. *ECCV Workshops*. 2018. Available online: <https://arxiv.org/abs/1809.00219> (accessed on 21 December 2022).
16. Zhang, W.; Liu, Y.; Dong, C.; Qiao, Y. RankSRGAN: Generative Adversarial Networks with Ranker for Image Super-Resolution. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3096–3105.
17. Chadha, A. iSeeBetter: Spatio-temporal video super-resolution using recurrent generative back-projection networks. *Computational Visual Media* **2020**, *6*, 307–317. [CrossRef]
18. Zhang, Z.; Sze, V. FAST: A Framework to Accelerate Super-Resolution Processing on Compressed Videos. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, Hawaii, USA, 21–26 July 2016; pp. 1015–1024.
19. Li, Y.; Jin, P.; Yang, F.; Liu, C.; Yang, M.; Milanfar, P. COMISR: Compression-Informed Video Super-Resolution. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 11–17 October 2021; pp. 2523–2532.
20. Chen, P.; Yang, W.; Wang, M.; Sun, L.; Hu, K.; Wang, S. Compressed Domain Deep Video Super-Resolution. *IEEE Trans. Image Process.* **2021**, *30*, 7156–7169. [CrossRef] [PubMed]
21. Zhang, H.; Zou, X.D.; Guo, J.; Yan, Y.; Xie, R.; Song, L. A Codec Information Assisted Framework for Efficient Compressed Video Super-Resolution. *Eur. Conf. Comput. Vis.* **2022**, 1–16.
22. ISO/IEC 14496-10 Advanced Video Coding. Available online: <https://www.iso.org/obp/ui/#iso:std:iso-iec:14496:-10:ed-9:v1:en> (accessed on 21 December 2022).
23. Liu, H.; Ruan, Z.; Zhao, P.; Shang, F.; Yang, L.; Liu, Y. Video super-resolution based on deep learning: A comprehensive survey. *Artif. Intell. Rev.* **2020**, *55*, 5981–6035. [CrossRef]
24. Huang, Y.; Wang, W.; Wang, L. Bidirectional Recurrent Convolutional Networks for Multi-Frame Super-Resolution. *NIPS* **2015**.
25. Zhang, J.; Xu, T.; Li, J.; Jiang, S.; Zhang, Y. Single-Image Super Resolution of Remote Sensing Images with Real-World Degradation Modeling. *Remote. Sens.* **2022**, *14*, 2895. [CrossRef]

26. Jo, Y.; Oh, S.W.; Kang, J.; Kim, S.J. Deep Video Super-Resolution Network Using Dynamic Upsampling Filters Without Explicit Motion Compensation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3224–3232.
27. Haris, M.; Shakhnarovich, G.; Ukita, N. Deep Back-Projection Networks for Super-Resolution. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1664–1673.
28. Shi, W.; Caballero, J.; Huszár, F.; Totz, J.; Aitken, A.P.; Bishop, R.; Rueckert, D.; Wang, Z. Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1874–1883.
29. Yi, P.; Wang, Z.; Jiang, K.; Jiang, J.; Lu, T.; Tian, X.; Ma, J. Omniscient Video Super-Resolution. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 4409–4418.
30. Kim, J.; Lee, J.K.; Lee, K.M. Deeply-Recursive Convolutional Network for Image Super-Resolution. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1637–1645.
31. Tao, X.; Gao, H.; Liao, R.; Wang, J.; Jia, J. Detail-Revealing Deep Video Super-Resolution. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice Italy, 22–29 October 2017; pp. 4482–4490.
32. Liu, D.; Wang, Z.; Fan, Y.; Liu, X.; Wang, Z.; Chang, S.; Huang, T.S. Robust Video Super-Resolution with Learned Temporal Dynamics. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2526–2534.
33. Li, F.; Bai, H.; Zhao, Y. Learning a Deep Dual Attention Network for Video Super-Resolution. *IEEE Trans. Image Process.* **2020**, *29*, 4474–4488. [[CrossRef](#)] [[PubMed](#)]
34. Fu, L.; Li, J.; Zhou, L.; Ma, Z.; Liu, S.; Lin, Z.; Prasad, M. Utilizing Information from Task-Independent Aspects via GAN-Assisted Knowledge Transfer. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018.
35. Zhang, L.; Li, J.; Huang, T.; Ma, Z.; Lin, Z.; Prasad, M. GAN2C: Information Completion GAN with Dual Consistency Constraints. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018.
36. Liu, R.; Wang, X.; Lu, H.; Wu, Z.; Fan, Q.; Li, S.; Jin, X. SCCGAN: Style and Characters Inpainting Based on CGAN. *Mob. Netw. Appl.* **2021**, *26*, 3–12. [[CrossRef](#)]
37. Mittal, A.; Soundararajan, R.; Bovik, A.C. Making a “Completely Blind” Image Quality Analyzer. *IEEE Signal Process. Lett.* **2013**, *20*, 209–212. [[CrossRef](#)]
38. Ma, C.; Yang, C.; Yang, X.; Yang, M. Learning a No-Reference Quality Metric for Single-Image Super-Resolution. *Comput. Vis. Image Underst.* **2016**, *158*, 1–16. [[CrossRef](#)]
39. Blau, Y.; Michaeli, T. The Perception-Distortion Tradeoff. *Computer Vision and Pattern Recognition*. 2017. Available online: <https://arxiv.org/abs/1711.06077> (accessed on 12 December 2022).
40. Qin, X.; Ban, Y.; Wu, P.; Yang, B.; Liu, S.; Yin, L.; Liu, M.; Zheng, W. Improved Image Fusion Method Based on Sparse Decomposition. *Electronics* **2022**, *11*, 2321. [[CrossRef](#)]
41. Liu, H.; Liu, M.; Li, D.; Zheng, W.; Yin, L.; Wang, R. Recent Advances in Pulse-Coupled Neural Networks with Applications in Image Processing. *Electronics* **2022**, *11*, 3264. [[CrossRef](#)]
42. Dong, C.; Li, Y.; Gong, H.; Chen, M.; Li, J.; Shen, Y.; Yang, M. A Survey of Natural Language Generation. *ACM Comput. Surv.* **2022**, *55*, 1–38. [[CrossRef](#)]
43. Zabalza, M.C.; Bernardini, A. Super-Resolution of Sentinel-2 Images Using a Spectral Attention Mechanism. *Remote. Sens.* **2022**, *14*, 2890. [[CrossRef](#)]
44. Zhang, R.; Isola, P.; Efros, A.A.; Shechtman, E.; Wang, O. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 586–595.
45. Sajjadi, M.S.; Schölkopf, B.; Hirsch, M. EnhanceNet: Single Image Super-Resolution Through Automated Texture Synthesis. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2016; pp. 4501–4510.
46. Johnson, J.; Alahi, A.; Fei-Fei, L. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. *arXiv* **2016**, arXiv:abs/1603.08155.
47. Huang, J.; Singh, A.; Ahuja, N. Single Image Super-Resolution from Transformed Self-Exemplars. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 5197–5206.
48. Clark, A.; Donahue, J.; Simonyan, K. Efficient Video Generation on Complex Datasets. *ArXiv* **2019**, arXiv:ArXiv:abs/1907.06571.
49. Dong, C.; Deng, Y.; Loy, C.C.; Tang, X. Compression Artifacts Reduction by a Deep Convolutional Network. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 576–584.
50. Kim, K.I.; Kwon, Y. Single-image super-resolution using sparse regression and natural image prior. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 1127–1133. [[PubMed](#)]
51. Timofte, R.; De Smet, V.; Van Gool, L. Anchored neighborhood regression for fast example-based super-resolution. In Proceedings of the IEEE International Conference on Computer Vision (ICCV); 2013; pp. 1920–1927.
52. H.264 Reference Software. Available online: <https://iphome.hhi.de/suehring/tml/download/> (accessed on 21 December 2022).

53. She, Q.; Hu, R.; Xu, J.; Liu, M.; Xu, K.; Huang, H. Learning High-DOF Reaching-and-Grasping via Dynamic Representation of Gripper-Object Interaction. *ACM Trans. Graph.* **2022**, *41*, 1–14. [[CrossRef](#)]
54. 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 Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2016; pp. 2848–2857.
55. ITS. Consumer Digital Video Library. Available online: <https://www.cdvl.org/> (accessed on 22 December 2022).
56. Ma, D.; Afonso, M.; Zhang, F.; Bull, D.R. Perceptually-inspired super-resolution of compressed videos. *Opt. Eng. + Appl.* **2019**, *11137*, 1113717.
57. Keys, R. Cubic convolution interpolation for digital image processing. *IEEE Trans. Acoust. Speech Signal Process.* **1981**, *29*, 1153–1160. [[CrossRef](#)]
58. Turkowski, K. Filters for Common Resampling Tasks. In *Graphics Gems*; Academic Press Professional, Inc.: Cambridge, MA, USA, 1990; pp. 147–165.
59. Mo, Y.; Chen, D.; Su, T. A lightweight hardware-efficient recurrent network for video super-resolution. *Electron. Lett.* **2022**, *58*, 699–701. [[CrossRef](#)]
60. Shang, F.; Liu, H.; Ma, W.; Liu, Y.; Jiao, L.; Shang, F.; Wang, L.; Zhou, Z. Lightweight Super-Resolution with Self-Calibrated Convolution for Panoramic Videos. *Sensors* **2022**, *23*, 392. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.