A Survey of Deep Learning Video Super-Resolution

Arbind Agrahari Baniya, *Graduate Student Member, IEEE*, Tsz-Kwan Lee, Peter W. Eklund,
and Sunil Aryal, *Member, IEEE*

***Abstract*—Video super-resolution (VSR) is a prominent research topic in low-level computer vision, where deep learning technologies have played a significant role. The rapid progress in deep learning and its applications in VSR has led to a proliferation of tools and techniques in the literature. However, the usage of these methods is often not adequately explained, and decisions are primarily driven by quantitative improvements. Given the significance of VSR's potential influence across multiple domains, it is imperative to conduct a comprehensive analysis of the elements and deep learning methodologies employed in VSR research. This methodical analysis will facilitate the informed development of models tailored to specific application needs. In this paper, we present an overarching overview of deep learning-based video super-resolution models, investigating each component and discussing its implications. Furthermore, we provide a synopsis of key components and technologies employed by state-of-the-art and earlier VSR models. By elucidating the underlying methodologies and categorising them systematically, we identified trends, requirements, and challenges in the domain. As a first-of-its-kind survey of deep learning-based VSR models, this work also establishes a multi-level taxonomy to guide current and future VSR research, enhancing the maturation and interpretation of VSR practices for various practical applications.**

***Index Terms*—Video super-resolution, deep learning, upsampling, fusion, survey, downsampling, alignment, loss function.**

I. Introduction

An INCREASE in the consumption of video-based multimedia in recent years can be attributed to advances in video capture technology, transmission networks, and rendering devices. These advances have led to an increased demand for higher-quality video signals. Quality can be defined from two standpoints, the quality of service (QoS) and quality of experience (QoE). From a QoS perspective, higher quality refers to a video bitstream with a higher bitrate, larger spatial resolution, and/or higher temporal resolution (more frames per second). While from a QoE perspective, higher quality is subjective and can be difficult to quantify as it aligns with a more pleasing perception, a judgement that varies greatly among individuals. It has been established that enhancements along the spatiotemporal dimensions of video signals result in an improved quality that positively correlates to an improvement in QoS and is, in turn, linked to perceived improvements in QoE [1]. The enhanced resolution improves various aspects of video quality and user experience. As a result, video super-resolution (VSR) models are widely being developed [2], [3], as the process of generating high-resolution (HR) video output with improved quality from a given low-resolution (LR) video input. Assuming a high-resolution video has undergone the following operation:

$$LR = (HR * k)\downarrow_d + ns \tag{1}$$

where LR is the low-resolution video when each frame of high-resolution video HR is convoluted with a blur or cubic kernel k followed by downsampling operation d and addition of noise ns, super-resolution of LR is then a task of estimating the kernel, downsampling operation, and the noise such that HR video can be obtained inversely from LR video. As implied by (1), VSR is an ill-posed inverse problem that is considered an open research area in the low-level computer vision domain. VSR has mostly been treated as an extension of single-image super-resolution (SISR) and multi-image super-resolution (MISR). However, unlike SISR and MISR, modelling the tasks in VSR is challenging due to the need to aggregate highly correlated but misaligned frames in a given video sequence [4], [5]. Adopting the approaches used in conventional SISR and MISR designed for an image directly to video-based super-resolution may fail to capture the temporal reliance between video frames [6], [7]. Therefore, recent studies have adopted learning-based approaches that exploit spatiotemporal features present in an LR video to super-resolve it to HR video [8], [9], [10], [11], [12], [13].

Traditionally, upsampling algorithms such as back-projection methods [14] and least mean squares (LMS) based Kalman filter methods [15] have been used to interpolate pixels in a video frame or a single image. These methods rely on deterministic functions to map LR input to HR output. However, the deterministic nature of traditional methods limits their capacity to generalise well over different videos, and the inverse function obtained by traditional methods does not capture the non-linear complexity of the transformation that maps HR to LR video. Consequently, deep learning VSR models have garnered significant attention in recent times owing to their stochastic and data-centric characteristics, enabling effective generalisation across diverse video inputs. Moreover, these models possess the capacity to learn non-linear functions for mapping LR videos to their HR counterparts. Learning-based methods for VSR typically comprise feature extraction, alignment, fusion, reconstruction,

Manuscript received 27 July 2023; revised 17 March 2024; accepted 22 April 2024. Date of publication 17 May 2024; date of current version 24 July 2024. The work of Sunil Aryal was supported by the US Air Force Office of Scientific Research (AFOSR) under Grant FA2386-20-1-4005. *(Corresponding author: Arbind Agrahari Baniya.)*

The authors are with the School of Information Technology, Deakin University, Geelong, Australia (e-mail: a.agraharibaniya@deakin.edu.au; glory.lee@deakin.edu.au; peter.eklund@deakin.edu.au; sunil.aryal@deakin.edu.au).

Recommended for acceptance by H. Yin.

Digital Object Identifier 10.1109/TETCI.2024.3398015

and up-sampling as fundamental steps in the super-resolution process. The extraction of relevant features from accurately aligned frames and their subsequent fusion is of utmost importance in such models [13], [16], [17]. In this paper, we thoroughly investigate each component of deep learning-based VSR models. To date, there has only been one work published in this space [18]; however, the complexity in the VSR domain is significantly diluted with a single-layer taxonomy focusing only on the alignment step. Several other components within VSR are remarkably diverse and thus contribute to increasingly varied outcomes that are often difficult to interpret and explain. This paper aims to bridge these gaps by:

- developing a novel taxonomy and extensive listing of approaches and trends within individual VSR components;
- providing a thorough methodological review of deep learning in the context of video super-resolution;
- providing a comprehensive overview of VSR literature, current trends, applications, and challenges;
- making the VSR models and their respective performances more explainable;
- and providing future VSR works with guidelines based on the prospective requirements and gaps.

II. Background

A. Image Super-Resolution (ISR)

Single image super-resolution (SISR) is a technique used to enhance the resolution of a single image by increasing the number of pixels in the image [19]. The goal of SISR is to generate a high-resolution image from a low-resolution counterpart by interpolating missing pixels and adding high-frequency details. At the same time, multi-image super-resolution (MISR) is a technique used to enhance the resolution of single or multiple images by combining the infusion of pixels from multiple images [20]. The earliest methods for image super-resolution were based on interpolation, such as nearest-neighbour [21] and bicubic interpolation. These methods are simple to implement and computationally efficient but could not exploit the high-frequency information present in the images, leading to artefacts and a lack of details. Example-based ISR methods were then proposed [22], such as self-exemplar and sparse-coding-based methods. These methods used external examples to guide the interpolation process, improving the quality of the generated HR images. However, they are limited by the quality and availability of the examples. Similarly, image prior-based super-resolution techniques exploited prior knowledge about the regularities and structures found in high-resolution images to super-resolve low-resolution inputs [23], [24]. The assumptions and restrictive adaptability of priors-based approaches result in limited applicability and generalisability while involving computationally complex optimisation algorithms.

Deep learning-based ISR methods, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), transformers and generative adversarial networks (GANs), have been gaining popularity [25], [26] because of their ability to learn complex and non-linear mappings from LR to HR images and generate high-quality HR outputs by exploiting the low-frequency information present in the input [27]. The success of these approaches is significantly attributed to the deep neural networks' ability to automatically extract features of interest and the hierarchical representational ability of the non-linear complex patterns required to be restored for super-resolution. More recently, efficiency concerning deep learning is leading ISR models to mitigate complex networks with high computational and memory demands. Lightweight CNN and transformer backbones are being used to significantly reduce memory usage compared to traditional transformers [28]. However, the trade-off between reconstruction performance and efficiency is to be balanced better. An example is the usage of hierarchical dense residual blocks incorporated without significantly increasing computational overheads [29]. Non-local operations and sparse representations are also being utilised to improve both quantitative metrics and visual quality [30]. Other alternatives tackling the computational demands in transformers incorporate shift convolution and group-wise multi-scale self-attention modules, offering superior performance with significantly reduced computational complexity [31]. Nevertheless, overcoming computational challenges while maintaining or surpassing state-of-the-art performance remains a focused trend in current ISR research [20].

B. From Image to Video Super-Resolution

The fundamental difference between images and video is that video comprises multiple frames over an added temporal dimension. This temporal dimension of video adds additional complexity to the super-resolution task, as it requires aligning and fusing multiple temporally dispersed frames to generate a high-resolution video. Extending the target-resolving subject from image to video signals, super-resolution approaches used in conventional ISR to VSR fail to capture the unique temporal information present in videos. VSR aims to adopt several temporally correlated low-resolution frames within a video sequence to super-resolve the frame series. Considering spatial and temporal dimensions across multiple input frames induces VSR as a highly non-linear multi-dimensional problem. VSR serves online and offline application contexts addressing three primary objectives: enhancing QoS, improving QoE, and assisting computer vision systems. It elevates the sharpness and detail of low-resolution videos in platforms like video streaming and multimedia communication, enriches the visual experience in entertainment and gaming, and aids accurate analysis and recognition in computer vision systems, essential for surveillance, autonomous vehicles, and robotics.

Attempts have been made to translate the ISR problem definition and solutions to VSR. For instance, many early VSR methods directly applied ISR methods, such as interpolation-based, example-based [32] and video prior-based [33] methods, to video frames without considering the temporal dimension of the video. However, these methods led to temporal inconsistencies and a lack of detail with a smoothing effect in the generated HR video [34]. Improving on this, VSR methods began incorporating additional techniques such as motion estimation

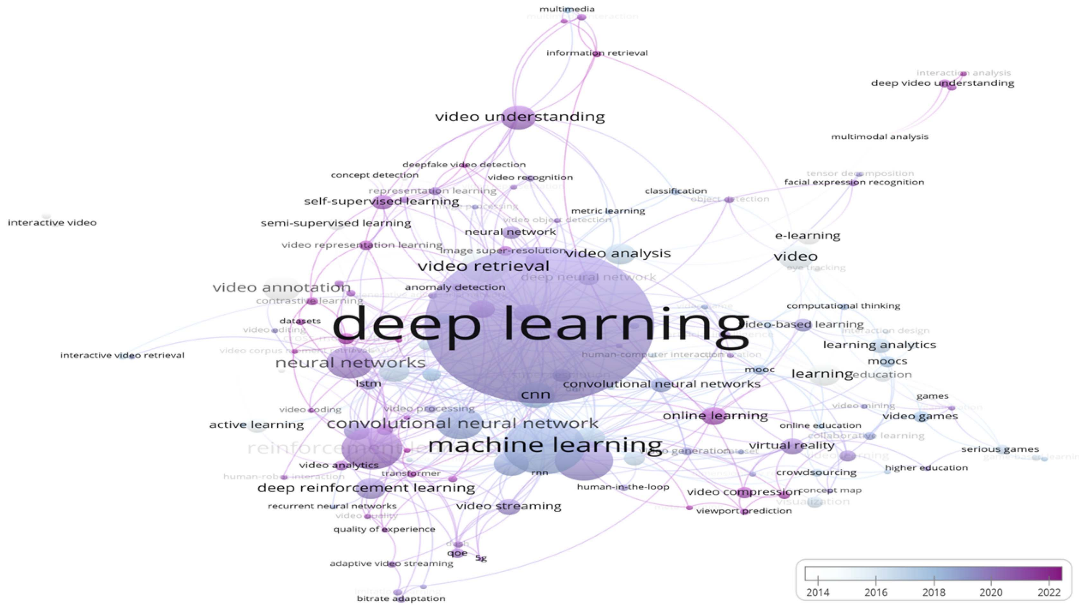

Fig. 1. Publication keyword cloud in last two decades with circle size representing number of occurrences and colour highlighting chronological occurrence.

and compensation (MEMC) and temporal fusion in addition to the conventional ISR methods to exploit the spatiotemporal correlation present in the video. This improved the quality of the generated HR video, resulting in a more realistic and natural output. Examples of such methods include traditional methods like Kalman filter-based methods [35], [36] and adaptive filtering methods [37], and more recently, deep learning-based methods [38].

Deep learning-based methods have been widely adopted in VSR due to their effectiveness in automatically extracting and learning desired features by leveraging the spatiotemporal information present in the video. The use of deep learning has enabled the development of end-to-end VSR models that can learn the mapping between LR and HR videos in a data-driven manner. Earlier deep-learning models tried to learn the mapping between LR and HR frames by extracting residual features using LR frame(s) and motion details [16]. These early models still lacked the sequential modelling ability desired to learn the true nature of videos. As a result, there is a growing interest in developing VSR models for exploiting temporal dependencies between video frames and capturing long-term spatiotemporal patterns across the time domain. One popular approach is to use recurrent neural networks (RNNs) with memory-preserving techniques such as residual blocks with skip connections and long-short term memory (LSTM) to model the temporal dynamics of the video. RNN-based VSR models effectively learn the underlying temporal patterns and capture the long-term dependencies between frames, resulting in temporally consistent reconstructed HR videos [13], [39], [40]. Another approach is to use CNNs with 3D convolutions to capture both spatial and temporal features in the video [41], [42]. These models can effectively learn the spatiotemporal patterns present in the video and provide a more accurate mapping between LR and HR frames. However, the extent of temporal context that 3D CNN-based models can learn is limited to a fixed temporal window, unlike the global information propagation in RNNs.

Furthermore, recent advancements in transformers using attention mechanisms have led to the development of attention-based VSR models. These transformers selectively focus on relevant frames and regions within the video, improving the reconstruction quality by attending to the most informative aspects of the video [9], [43]. These attention-based VSR models effectively capture the spatiotemporal patterns in the video and have significantly advanced the state-of-the-art by improving the reconstruction quality of LR videos. However, as the field progresses, increasingly employing learning technologies to model the task of VSR, there are many unexplored sequential modelling advances for further investigation. To analyse the trend of work being done in VSR using deep learning, we explore the two popular databases, namely IEEE Xplore, with a search term ((“All Metadata”: ‘video super-resolution’) AND (“All Metadata”: ‘deep learning’)) and ACM Digital Library with a search term ([Title: ‘video super-resolution’] AND [Title: ‘deep learning’]) for works published in the past two decades. We collect a total of 1,108 published works and plot a yearly trend as shown in Fig. 2. The figure demonstrates a steadily increasing trend, particularly escalating in the last decade. We further extract the keywords from these publications and plot a keyword map showcasing the occurrences of keywords and relationships as shown in Fig. 1. Evident here (among other things) is the greater use and frequency of “deep learning” and “convolutional neural networks” reinforcing more recent and popular key terms.

C. Super-Resolution of Emerging Video Formats

1) Omnidirectional Video Super-Resolution: Omnidirectional videos, also known as 360° videos, capture the complete

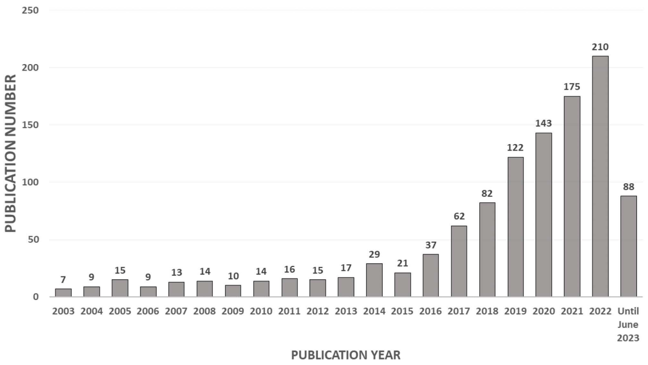

Fig. 2. The number of publications in deep learning for video super-resolution in the last two decades by year.

360° horizontal view of the surrounding environment and provide an immersive viewing experience. These videos are primarily used in virtual reality (VR) and augmented reality (AR) to provide immersive experiences as part of an extended reality. Omnidirectional video super-resolution aims to enhance the resolution of these videos to improve the visual quality and details it to improve the immersivity [44]. The unique characteristics of omnidirectional videos, such as the equirectangular projection, pose challenges for traditional video super-resolution methods. The spatially varying distortion in the equirectangular projection and the need to handle 360° coverage requires specialised techniques for omnidirectional video super-resolution. Despite potentially being an obvious extension of conventional VSR, the number of works done in the omnidirectional video super-resolution literature is limited [18], [45]. Dasari et al. [46] propose a micro-model for super-resolution in 360° videos to address bandwidth-related requirements for adaptive video streaming. Their approach focused on enhancing the spatial quality of compressed tiles by passing them through multiple convolution layers and final deconvolution and upsampling. Liu et al. [47] introduced a dual network VSR model, named Single and Multi-Frame Recurrent Network (SMFN), for 360° videos. One pipeline handles SISR, and the other handles MISR. SMFN has shown better results than conventional VSR models such as EDVR [11] and RBPN [16], with targeted training on a 360° video dataset. More recently, Agrahari Baniya et al. [45] proposed a spherical signal super-resolution with a proportioned optimisation (S3PO) using recurrent modelling and alignment-free 360° feature extraction surpassing state-of-the-art VSR models like BasicVSR [13] in super-resolving omnidirectional videos. Although it has been shown that direct adoption of conventional 2D alignment [48] and other VSR components into 360° VSR models is not the answer to super resolving omnidirectional videos, more work needs to be done to unpack the applicability of 2D deep learning technologies for 360° VSR problem.

2) 3D Video Super-Resolution: 3D video super-resolution aims to enhance the resolution of videos captured with depth information and/or multi-view cameras. By leveraging the additional depth, or multi-view information, 3D video super-resolution techniques generate high-resolution videos with improved spatiotemporal details as well as better depth accuracy [49]. Traditional 2D video super-resolution methods are limited in their ability to handle depth information or multi-view data. 3D video super-resolution methods consider both the spatiotemporal and depth dimensions of the video to perform super-resolution [50]. These methods need to exploit the geometric correlations between different views or depth maps in a low-resolution video to generate a high-resolution counterpart. Different approaches have been explored in 3D video super-resolution. These include depth-based methods that utilise depth maps to guide the super-resolution process [51], view synthesis-based methods that synthesise high-resolution views from the available low-resolution views, and fusion-based methods that combine information from multiple views to generate a high-resolution video [52]. With added challenges of occlusions, large disparities between views and depth maps, and limited comprehensive benchmark datasets to be used for model training and development, 3D video super-resolution is, therefore, an emerging field of study.

Although it is a relatively mature field of research, 2D video super-resolution remains an active research area. Numerous unresolved challenges in sequential modelling, along with the emergence of new requirements/applications, have meant that VSR is a progressive and continually evolving research topic. However, often the research conducted in this area lacks the consideration of sequential modelling required for videos and the applicability of these models in terms of the data required as input, model sizes and inference efficiencies. On top of this, as with many other domains implementing deep learning techniques, the composite impact and ability of deep learning methodologies in relation to objective and subjective outcomes remain unexplained. The following sections of this paper aim to investigate the key components and aspects related to the VSR technologies being used, aimed to guide informed model development with explainable methodological choices.

III. Overview of Deep Learning-Based VSR

VSR models employing deep learning predominantly make use of five fundamental methodological components, namely - input, alignment, fusion, refinement and upsampling. Each of these components has a diverse range of methodological options with specific purposes and implications. Fig. 3 provides a taxonomic categorisation of the most commonly used options under each component across VSR stages. This taxonomy forms the baseline for the study of literature and methodological analysis conducted in this paper. The technical details, purpose and corresponding substances of each component are discussed in detail in the following sections.

A. *Inputs*

One of the most important aspects of VSR is the input, as the quality and content of the input directly affect the learning ability of data-driven VSR models and the corresponding outputs generated. In this section, we discuss techniques used for obtaining the input in VSR, including synthetic downgrading and downsampling and real-world LR inputs and how these are

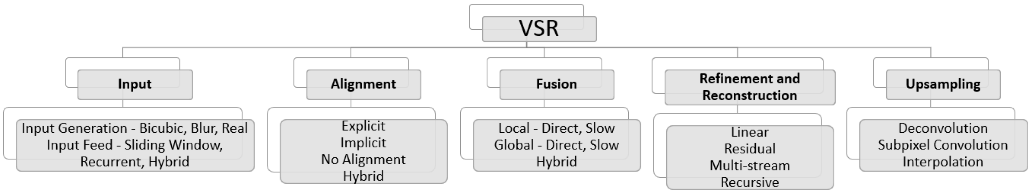

Fig. 3. Taxonomy for components across various stages in a VSR model based on the detailed discussion in Section III.

fed to VSR models using either a sliding window, in a recurrent fashion, or via a combination of both.

1) Input Generation: Synthetic downgrading is a common technique to create LR frames from HR frames for training and evaluating VSR models. This is done by applying various types of quality degradation to the HR frames, such as downsampling, blurring, and the introduction of noise.

- Bicubic downsampling is a simple and widely used technique for creating LR frames from HR frames. Bicubic downsampling involves downsampling the HR frame by a factor of d and applying a cubic interpolation filter across the frame. The filter is typically a square kernel of length $4d + 1$ where d is the downsampling factor. Downsampling is obtained by taking a weighted average of every d pixels to obtain $\times d$ reduced spatial resolution. Bicubic downsampling leads to the loss of high-frequency information, reduced spatial pixels and artifacts, such as ringing near edges, namely unwanted oscillations or ripples around sharp edges, resulting in a distorted appearance. The degraded LR image is then used to train deep VSR models in conjunction with its HR ground truth in a supervised manner.
- Blur or Gaussian downsampling is another technique used to create LR frames from HR frames. Blur or Gaussian downsampling involves convolving the HR frame with a Gaussian kernel (G_σ) with standard deviation σ, which effectively blurs the frame and reduces its granular details. The resulting frame is then downsampled by sampling one pixel out of every d pixel to obtain $\times d$ reduced spatial resolution. Compared to bicubic downsampling, aliased artifacts are relatively fewer, and more high-frequency information is preserved when downsampling with a Gaussian filter. This means that restoring high-frequency details from bicubically downsampled low-resolution video is a more challenging task to learn for deep VSR models compared to that with blur downsampling. Moreover, bicubic downsampling is computationally efficient and requires less processing time compared to Gaussian/blur downsampling since blur downsampling involves applying a computationally intensive convolution operation.
- Real-world LR inputs refer to frames that have been degraded in the real world or closely imitated, including frames captured by low-quality cameras, frames that have been lossy compressed, and frames that have been resized or down-sampled randomly with unknown parameters. A few popular approaches are to apply dynamic blur and image compression techniques, such as JPEG or video compression techniques with random codecs and bitrates. These frames have the advantage of being more representative of the types of degraded frames that users encounter in the real world and, thus, contain degradation that is challenging to revert. Additionally, diversity in degradation can make it challenging to develop a single VSR algorithm that works well for all types of real-world LR videos. Often referred to as blind super-resolution [6], [53], [54] or real super-resolution [55], the research area aiming to super-resolve real-world low-resolution videos to a high-resolution counterpart remains in its infancy.

2) Input Feed:

- Sliding-window Feed — The sliding window approach is a simple and widely used method for inputting frames into a VSR model. It involves dividing a video into overlapping segments of a fixed size and then feeding each segment, also called a "window" into the VSR model. The window size is chosen to be small enough to capture temporal variations in the video while also being large enough to provide enough temporal context for the VSR model to learn spatiotemporal correlations. One advantage of the sliding window approach is that it allows the VSR algorithm to use information from past and future frames to generate the output. This can be especially beneficial in cases with significant motion or luminance changes between consecutive frames since the VSR model can establish the spatiotemporal context from other neighbouring frames to better estimate the high-resolution output. The overlapping segments allow flexibility in the input size. While the optimal number of frames per sliding window is subjective to the model and video dataset, limited works have attempted to investigate the effect of the number of frames used in the sliding window input. Increasing the number of frames from three to five in a sliding window results in improved super-resolution performance; however, increasing the window size beyond five results in consequential computational overhead that eclipses any further result improvement [16].
The sliding window approach is computationally expensive as the VSR model must process each frame multiple times due to the overlapping windows. The overlapping segments may also introduce and propagate repetitive noise into the input across several timestamps, which can negatively impact the quality of the output across the frame series. The number of frames required per target frame resolution has decreased in recent models, with improved temporal information propagation abilities mitigating the need to

TABLE I
SUMMARY OF VSR MODELS AND THE COMPONENTS USED WITHIN THE RESPECTIVE MODEL BASED ON THE TAXONOMY IN FIG. 3

Model	Year	Input Generation	Input Feed	Alignment	Fusion	Refinement
VSRnet [57]	2016	Bicubic	Sliding Window	Explicit (MEMC)	Local (Direct)	Linear
VESPCN [58]	2017	Bicubic	Sliding Window	Explicit (MEMC)	Local (Direct)	Linear
SPMC [59]	2017	Bicubic	Sliding Window	Explicit (MEMC)	Local (Direct)	Residual
BRCN [60]	2018	Blur, Bicubic	Recurrent	No Alignment	Global (Direct)	Linear
FRVSR [61]	2018	Blur	Recurrent	Hybrid	Global (Direct)	Residual
DUF [41]	2018	Bicubic	Sliding Window	No Alignment	Local(Direct)	Multistream, Residual
FSTRN [62]	2019	Bicubic	Sliding Window	No Alignment	Local (Direct)	Residual
3DSRnet [42]	2019	Bicubic	Sliding Window	No Alignment	Local (Direct)	Linear
TecoGAN []	2019	Bicubic	Recurrent	Hybrid	Global (Direct)	Recursive
RBPN [16]	2019	Bicubic	Sliding Window	Explicit (MEMC)	Local (Direct)	Residual
EDVR [11]	2019	Bicubic	Sliding Window	Implicit (Deformable)	Local (Direct)	Residual
RLSP [63]	2019	Blur	Recurrent	No Alignment	Global (Direct)	Linear
TDAN [64]	2020	Blur, Bicubic	Sliding Window	Implicit (Deformable)	Local (Direct)	Linear
TGA [9]	2020	Blur	Sliding Window	Explicit(MEMC)	Local (Slow)	Linear
RSDN [8]	2020	Bicubic	Recurrent	No Alignment	Global (Direct)	Residual, Multistream
RRN [39]	2020	Bicubic	Recurrent	No Alignment	Global (Direct)	Residual
MuCAN [65]	2020		Sliding Window	No Alignment	Local (Slow)	Linear
D3D [66]	2020	Bicubic	Sliding Window	Implicit (Deformable)	Local (Direct)	Residual
MSFFN [67]	2021	Bicubic	Recurrent	Implicit (Deformable)	Global (Slow)	Residual
STMN [68]	2021	Bicubic	Sliding Window	No Alignment	Global (Direct)	Residual
EVSRNet [69]	2021	Bicubic	Sliding Window	No Alignment	Local (Direct)	Residual
FDAN [70]	2021	Blur	Sliding Window	Implicit (Deformable) Explicit (Flow)	Local (Direct)	Residual
BasicVSR [13]	2021	Blur, Bicubic	Recurrent	No Alignment	Global (Direct)	Residual
IconVSR [13]	2021	Blur, Bicubic	Hybrid	Hybrid	Hybrid (Direct)	Residual
GOVSR [71]	2021	Blur	Hybrid	No	Hybrid (Direct)	Residual
MSHPFNL [72]	2022	Blur, Bicubic	Sliding Window	No Alignment	Local (Progressive)	Residual
TTVSR [73]	2022	Blur, Bicubic	All Frames	Explicit (MEMC)	Global (Direct)	Residual
BasicVSR++ [74]	2022	Blur, Bicubic	Recurrent	Hybrid	Global (Direct)	Residual
PSRT [75]	2022	Bicubic	Sliding Window	Explicit (MEMC), Implicit (Deformable)	Local (Direct)	Residual
R2D2 [40]	2023	Blur	Hybrid	Hybrid	Hybrid (Direct)	Residual, Multistream
R2D2-*lite* [40]	2023	Blur	Hybrid	No	Hybrid (Direct)	Residual, Multistream

process videos as disparate pockets of correlated information. However, the temporal sliding window remains the most commonly used input feed mechanism in VSR literature, as observed in Table I. Despite the popularity of this approach, little is done to treat the temporally dispersed frames in a sliding window differently in relation to the target frame being super-resolved [11]. Attempts have been made to further segment the sliding window into smaller segments of temporally co-located frames, in either temporal direction with reference to the target frame, and apply different dilations based on temporal distance [9]; however, this approach still assumes a fixed correlation based on temporal distance instead of the actual spatiotemporal dynamics present within each video. Agrahari Baniya et al. [56] have recently demonstrated an alternative sliding-window mechanism with selection measures integrated to perform suitability selection in the input space. Pixel-based and feature-based selections, in this case, have proven to improve the VSR performance showing a direct impact and encouraging consideration of spatiotemporal dynamics in a sliding window for VSR.

- Recurrent Feed — The recurrent window approach is a more recent method for inputting frames into a VSR algorithm. Recurrent Feed is used with RNNs to feed frames to the VSR model in a recurring manner in a way that is meaningful when used in conjunction with global memory propagation. Depending on the nature of the RNN model, the recurring window can have either unidirectional or bidirectional feeds. The Recurrent feed method usually incorporates the target frame to be super-resolved in confluence with the global memory propagated from the distant timestamps. A simple change in input feed with the use of recurrent feedback has shown to be effective and produce competitive results, often superior [39] to using sliding window frames. This embodies the importance of modelling the task of VSR as a sequential modelling task with recurrent input feed.
- Hybrid Feed — This approach combines elements of both the sliding window and recurrent approaches. Hybrid Feed involves dividing the video into overlapping segments in a similar way to the sliding window approach, but instead of processing each segment independently, the VSR algorithm maintains a recurrent memory state that captures the temporal context across segments. By incorporating both spatial and temporal information, the hybrid approach aims to leverage the benefits of both sliding window and recurrent methods. The hybrid feed allows the VSR algorithm to utilise local spatiotemporal context from previous and future proximal frames within each window while also considering the temporal dependencies propagated across the global timestamps [13]. This has led to improved performance in capturing both short-term and long-term spatiotemporal patterns and producing high-quality super-resolved videos. Ideally, the number of frames per input in

a hybrid feed is usually an additional past and/or future frame in reference to the target frame along with the globally propagated recurrent memory [8], [40]. Unlike the sliding window method, the need for a large temporal radius for sliding windows is mitigated by global memory propagation. However, the usage of recurrent and hybrid feeds is only applicable to recurrent VSR networks that can maintain recurrent feedback propagation.

B. Alignment

Alignment refers to the process of aligning low-resolution neighbouring frames or features with their corresponding target counterparts being super-resolved. There are several types of frame/feature alignment methods used in VSR, including:

1) Explicit Alignment: Motion Estimation and Motion Compensation (MEMC) based alignment is a widely used method for aligning frames and features explicitly in VSR. The method uses dense optical flow estimated between neighbouring and target frames/features to align them. It computes the motion vector for each pixel by analysing the changes between consecutive frames, assuming that the intensity pattern observed in a frame remains constant over time and any variation is due to the movement of objects or the camera. The flows are then used to warp the neighbouring frames/features to match the position of the target counterpart. This method is efficient and has been widely used [58]. However, it has some limitations when dealing with complex, fast, and large motions or significant changes in luminance. In these cases, the estimated optical flow may not be accurate, leading to misalignment and degraded VSR performance. To mitigate some of the limitations, traditional methods of flow estimations are being increasingly replaced with learning-based motion estimation methods [76], such methods proving to be more robust to various motion types.

2) Implicit Alignment: Deformable convolution [77] is a method that allows convolutional kernels to adapt to the shape of the input feature map. This method has been applied to align frames and features implicitly in VSR and has shown promising results [78]. Deformable convolution learns the geometric variations and motion of objects in the video and deforms the kernel to adjust it accordingly. The method thus provides implicit alignment capabilities. Deformable convolution is robust to changes in motion and luminance; however, it requires a large number of parameters and thus increases the computational overhead of the VSR model.

3) No Alignment: Instead of aligning frames, features are extracted directly from frames using either 2D, 3D or recurrent neural networks without any alignment. These methods have the advantage of being computationally efficient and do not require any explicit or implicit alignment overhead [61]. However, no alignment as a strategy may not be as effective as alignment-based methods in handling complex motions since the VSR model solely relies on the effectiveness of spatial or spatiotemporal features extracted by the 2D or 3D/recurrent convolution layers, respectively. Without the alignment to guide temporal coherence in model outputs, non-alignment methods are prone to inconsistencies, with the notable exception of RNNs, which maintain global memory propagations aimed at assisting temporal modelling. However, unwanted noises or changes in the recurrent feedback due to sudden large motion or luminance change are likely to propagate across a longer temporal radius if not interrupted and rectified. In addition to the use of 2D, 3D and recurrent networks for no alignment, non-local operations using Gaussian functions and dot products, pixel operations like RGB difference maps and forward and backward feature differences are used for finding relationships and correlations between frame pairs or feature pairs [72], [79], [80]. However, this approach introduces additional computational costs and is more often than not outperformed by recurrent methods.

4) Hybrid Alignment: More recently [13], [71], to mitigate the shortcomings of the no-alignment approach while leveraging its efficiency, hybrid methods combine alignment and non-alignment techniques to handle different types of motion and improve the accuracy of alignment. These methods provide VSR models with the flexibility to leverage the robustness of implicit/explicit alignment methods in the event of large or sudden motion changes while benefiting from the efficiency of non-alignment components when motion or luminance changes are less significant. An example is the patch alignment technique in transformers which uses image alignment, feature alignment and deformable convolution to align image patches instead of pixels to mitigate the computational overhead of pixel alignment and prevent the impact of alignment on attention capabilities [75]. Similarly, a memory-augmented attention module has been used to maintain a global memory of the entire training set learned as parameters of the network while using regular non-local attention to query current frame features in the global memory bank to memorise and utilise general video details during the super-resolution training [81]. By combining the benefits of different alignment methods, hybrid approaches aim to enhance the overall performance of VSR models. This has only been introduced in recent years with only reported usage in recurrent models using hybrid input feeds discussed earlier [40].

C. Fusion

Several types of feature/frame fusion are used in VSR models for integration or combination of multiple low-resolution frames or features before refining, reconstruction, and upsampling. Some of the most common methods include:

1) Local Information Fusion: Local fusion focuses on utilising the information from the immediate surrounding pixels or frames/features in a local neighbourhood to enhance the resolution of a specific area or a specific frame. These methods typically involve the use of convolution layers to extract features from the low-resolution (LR) frame(s)/feature(s) and use the subsequent features for further processes.

- Direct Fusion: Direct fusion methods involve concatenating the target LR input frame/feature with the features extracted from a local neighbourhood and then passing the concatenated features through a series of layers for further refinement, reconstruction and upsampling. In order to fuse the information, either concatenation along the depth (most common) or addition along the co-located spatial dimension is performed. Direct fusion results in the abstraction of the temporal variations and dilutes the correlation differences that might be preset in the information. The expectation is for deep model layers to be able to extract meaningful features from the fused information. Despite its simplicity, this approach has been most commonly used, as shown in Table I, and has proven to be effective with the ability of deep learning to automatically extract features of interest in relation to the super-resolution task.

- Slow Fusion: Slow fusion methods involve using multiple stages of feature extraction and fusion in series, where each convolution layer extracts features from the previous layer and fuses it with other relevant information. This approach allows for a more gradual and fine-grained fusion of the LR input with the extracted features allowing for a better hierarchical feature representation, which can result in improved HR output. Although fusions from 3D models and elongated progressive fusions are categorised differently to slow fusion in some of the literature [72], [79], given the nature of fusion irrespective of the number of steps, or type of convolution involved, we categorise multi-stage fusion methods under slow fusion for simplicity's sake. The slow fusion methods also make use of either concatenation or aggregation operations to fuse the information (frame(s) and/or feature(s)) gradually. Unlike direct fusion, this approach aims to overcome the dilution and abstraction of spatiotemporal variations in the information and establish gradual relationships between different sets of information. This is particularly beneficial for methods using sliding window feed since it helps establish a hierarchical relationship between the temporally dispersed frames within a sliding window. The direct cost of the additional computation in relation to the magnitude of the improved result is, however, not fully studied in the literature.

2) Global Information Fusion: Global information fusion focuses on utilising information from a larger temporal dimension to enhance the resolution of a specific frame. This class of methods typically involve the use of recurrent feedback to extract features from a wider video frame sequence and then use these features to generate the HR output. This fusion technique also makes use of either concatenation or aggregation to fuse the local frame(s) and/or feature(s) with global memory. The process of fusion can either be direct or slow, in a similar way to the local fusion methods. However, unlike the local approach, the features extracted will represent the temporal context across the frame sequence allowing for better sequential modelling resulting in temporal coherence. This approach is only used with recurrent models making use of a recurrent input feed with no alignment discussed earlier.

3) Hybrid Information Fusion: Hybrid information fusion methods combine both local and global information to take advantage of the strengths of both approaches. Different frequency details and spatiotemporal relevance can be presented by combining the two approaches [40]. This allows for a more comprehensive fusion strategy that can capture both local details and global temporal coherence. The global and local information, in this case, is fused either directly or slowly using concatenation or aggregation. The most common approach is to use hybrid fusion with earlier discussed hybrid input feed and the hybrid alignment techniques. For example, two sets of information obtained from hybrid alignment (explicit alignment in a sliding window and no alignment in recurring feed) can be fused directly (or slowly) over multiple layers.

D. Refinement and Reconstruction

1) Linear: Linear refinement is a process that entails utilising a solitary linear pipeline consisting of convolution layers to extract valuable features from the amalgamated input features. This method of refinement and reconstruction is simple but also limited in effectiveness. Each future layer depends on features propagated from previous layers, unable to mitigate any noisy or irrelevant features from being propagated, leading to a direct impact on the generated output. Moreover, increasing the depth of the neural models leads to the fading of information propagated during backpropagation which hinders the learning ability of these models. Linear refinement models are increasingly uncommon as the advancements in deep learning space continue to be accelerated.

2) Residual: Residual refinement involves using residual blocks with skip connections for propagation. Residual blocks are designed to allow the network to propagate extracted features in conjunction with input to the respective layers to mitigate the shortcomings of its linear counterparts. This approach specifically helps mitigate the propagation of noise and unwanted information induced at any layer while preventing backpropagation info from fading and reaching deeper layers. The residual approach is the most common refinement approach currently in VSR literature for the reconstruction of fused features, as is evident in Table I.

3) Multi-Stream: Multi-stream refinement uses multiple pipelines to propagate and refine features. Each pipeline can learn differently to extract different features from the input, and the output from each pipeline is combined to generate the final output. The individual pipelines within the multi-stream refinement approach can be linear or residual. This approach allows the network to learn more complex feature representation with dedicated pipelines; however, it can significantly increase the model complexity and size. This approach is less common in the literature, but it has been shown to be effective when used strategically. Recent evident usage of this approach was observed in RSDN [8] and R2D2 [40], where the two pipelines were used to propagate structure and detail information and local and global information, respectively. The effectiveness of this approach over a single pipeline of residual blocks was highlighted in both works.

4) Recursive: Recursive refinement is used to refine the fused features in a recursive manner by iteratively applying refinement steps. This approach involves repeatedly passing the fused features through refinement modules to improve the quality of the final feature gradually. Each refinement module can be designed using linear or residual recurrent architectures. Recursive refinement allows for iterative fine-tuning of the features, enhancing their representation and reducing artifacts. However, this approach can be computationally expensive due to the repeated refinement steps and may therefore require careful tuning to avoid over-fitting or convergence issues. Recursive refinement is very uncommon because of its repetitive nature and is often replaced with linear or residual counterparts used in conjunction with a recurrent neural network instead of recursively refining the features for every timestamp.

E. Upsampling

Upsampling refers to the process of increasing the spatial resolution of low-resolution feature(s) or frames to obtain high-resolution counterparts. Various methods are used for upsampling, including the following:

1) Transposed Convolution (Deconvolution): Transposed convolution, also known as deconvolution, is a popular method for upsampling in deep learning that uses learned kernels to upsample the feature maps by convolving them with the input [41], [82]. The learnable kernels are similar to convolutional kernels but with the dimensions reversed. During the deconvolution process, the kernel slides over the input feature map, and instead of performing a dot product as in standard convolution, it computes the outer product of the kernel with the input. This process effectively spreads the activations and increases the resolution of the feature map. Deconvolution layers often include parameters such as stride and padding, similar to convolutional layers. The stride determines the step size of the kernel during sliding, and the padding controls the size of the output feature map. By adjusting these parameters, the resolution of the output feature map can be controlled. However, deconvolution has a tendency to generate checkerboard-like artifacts in the output. It refers to a visual artifact that can occur when performing upsampling or transposed convolution operations which are characterised by an exhibition of a pattern resembling a checkerboard, with alternating square regions of high and low intensity, negatively impacting the visual quality of the super-resolved video. To mitigate this issue, various regularisation techniques, such as adjusting the stride or dilation rate, or using skip connections, can be employed. Stride refers to the step size or the distance by which a filter/kernel moves across an input image during convolution or pooling operations. It determines the amount of overlap or spacing between successive applications of the filter. While dilation, also known as atrous convolution, is a technique used to increase the receptive field of a convolutional layer without increasing the number of parameters. The receptive field refers to the region of the input that a neuron in the convolutional layer "sees" when performing convolution. In a regular convolution operation, each filter/kernel is applied to a local region of the input data, and the size of the output feature map is determined by the size of the filter and the stride used. The receptive field of a neuron in the convolutional layer depends on the filter size and the number of layers it has passed through in the network. With dilation, gaps or spaces are introduced between the kernel elements, effectively increasing the stride of the convolution. This results in an expanded receptive field without altering the size of the filter. The dilation rate controls the spacing between the elements of the kernel. Similarly, skip connection refers to a mechanism that allows the direct flow of information from one layer to another, bypassing intermediate layers in a deep neural network architecture.

2) Pixel Shuffle (Subpixel Convolution): Pixel shuffle, or subpixel convolution, is a technique for upscaling or increasing the spatial resolution that rearranges the elements of low-resolution feature maps to form high-resolution feature maps by performing depth-to-space transformations [83]. The rearrangement takes place in the channel dimension (depth) of the feature map. For example, a feature map with dimensions [batch size, channels, height, width], where "channels" denotes the number of feature channels, and "height" and "width" represent the spatial dimensions, is reshaped so that each channel is divided into smaller subgroups of d^2 elements where d is the upsampling factor. These subgroups are spatially rearranged to form a higher-resolution feature map. Specifically, the d^2 elements from each subgroup are combined together to form individual pixels in the output feature map, which has a higher resolution. This method does not introduce any additional learnable parameters, which in turn contributes to its computational efficiency and popularity in super-resolution models with effective results.

3) Bilinear or Bicubic Interpolation: Bilinear or bicubic interpolation methods are simple and computationally efficient upsampling techniques. These methods estimate the values of new pixels based on the surrounding pixels in the low-resolution frames. The filter kernels utilised in traditional bilinear or bicubic upsampling are static and unchangeable, with the only adjustment being the kernel's position based on the location of the newly generated pixel in the upsampled frame. When performing $\times 4$ upsampling, a fixed set of 16 kernels is employed in these conventional methods. While the kernels are fast, they rarely fully restore sharpness and preserve fine textures in the resulting frame regions. Bilinear or bicubic interpolation methods are often used in conjunction with residual features from the deep VSR models. Instead of relying on the deep model to create every pixel, the residual approach allows the model to focus on the details that can be added on top of either the bilinear or bicubic interpolated low-resolution target frame. This method remains the most common approach to upsampling. However, the upsampling of the residual feature, in this case, is usually done with the pixel shuffle operation.

IV. Architecture, Training and Datasets

A. Network Architecture

The deep network architecture plays a crucial role in enhancing the learning capability and determining the components employed in VSR models. It also governs the overall modelling approach for video super-resolution. Depending on the chosen architecture, VSR models can learn to super-resolve videos either sequentially or non-sequentially. In this section, we delve into the prevalent deep learning architectures frequently utilised in the development of VSR models, as outlined in the provided Table II.

1) Non-Sequential:

- *2-Dimensional Convolutional Neural Networks (2D CNNs)* have been the cornerstone of image-related tasks

TABLE II

COMPREHENSIVE SUMMARY OF KEY LEARNING ASPECTS OF VSR MODELS IN THE LITERATURE AND THEIR REPORTED OBJECTIVE PERFORMANCE ON DIFFERENT TEST DATASETS

Model	Network Arch.	Loss Function	Training Dataset	Test Dataset	PSNR/SSIM	Size	Application
VSRnet [57]	2D CNN	Euclidian Distance	Myanmar	Vid4 (Y)	24.84 / 0.7049	0.27	Online
VESPCN [58]	2D CNN	Weighted (L2, Huber)	CDVL	Vid4 (Y)	25.35/0.7557	0.88	Online
DRVSR [59]	Encoder-Decoder	Weighted (Warping + Euclidean)	SPMCS	SPMCS (Y)	29.89 / 0.84	2.17	Online
BRCN [60]	3D-Bidirec. RNN	L2 loss	YUV25	Vid4 (RGB)	24.43/0.6334	-	Offline
FRVSR [61]	RNN	Weighted (MSE, Warping Error)	Custom (vimeo.com)	Vid4 (Y)	26.69/0.8220	2.81	Online
DUF [41]	3D CNN	Huber loss	Custom	Vid4 (Y)	27.34 / 0.8327	5.82	Online
FSTRN [62]	3D CNN	Charbonnier loss	YUV25	-	-	-	Online
3DSRnet [42]	3D CNN	L2 Loss	Custom - smallSet Custom - largeSet	Vid4 (RGB) Vid4 (RGB)	25.46/0.7498 25.71/0.7588	0.11	Online
RBPN [16]	Encoder-Decoder	L1 loss	Vimeo-90k	Vid4 (Y) SPMCS-11 (Y) Vimeo-90k (Y)	27.12/0.818 30.10/0.874 37.16/0.9566	12.2	Online
EDVR [11]	2D CNN	Charbonnier loss	REDS Vimeo-90k Vimeo-90k	REDS4(Y) Vid4 (Y) Vid4 (RGB)	31.09/0.8800 27.35/0.8264 25.83/0.8077	20.60	Online
RLSP [63]	RNN	MSE	Vimeo-90k	Vid4 (Y)	27.55/0.838	4.21	Online
TDAN [64]	2D CNN	Weighted (Alignment MSE + Supre-resolution MSE)	Vimeo-90k	Vid4(RGB) SPMCS-30 (RGB)	26.42/0.789 30.38/0.854	1.97	Online
TGA [9]	2D and 3D CNN	L1 loss	Vimeo-90k	Vid4 (Y) Vid4 (RGB)	27.59/0.8419 26.10/0.8254	5.8	Online
RSDN [8]	RNN	Weighted (Charbonnier	Vimeo-90k + L2)	Vid4 (Y) Vid4 (RGB)	27.92/0.8505 26.43/0.8349	6.19	Online
RRN [39]	RNN	L1 loss	Vimeo-90k	Vid4 (RGB) SPMCS (RGB) UDM10 (RGB)	26.16/0.8209 28.28/0.8690 37.03/0.9534	3.4	Online
MuCAN [65]	Encoder-Decoder	Weighted (Charbonnier + Edge Aware Loss)	REDS Vimeo-90k	REDS Vimeo-90k (RGB)	30.88/0.8750 35.49/0.9344	-	Online
D3Dnet [66]	3D CNN	L2 loss	Vimeo-90k	Vid4 (RGB) Vimeo-90k (RGB) SPMC-11 (RGB)	26.52/0.799 35.65/0.9330 28.78/0.8510	2.58	Online
MSFFN [67]	RNN	Charbonnier loss	Vimeo-90k	Vid4 (Y) Vimeo-90k (Y) SPMC-11 (Y)	27.23/0.8218 37.33/0.9467 30.13/0.8769	8.5	Online
STMN [68]	3D CNN	MSE	Custom (Vimeo.com + 699pic.com)	Vid4(Y)	25.90/0.7878	-	Online
EVSRNet [69]	2D CNN	L1 Loss	REDS	REDS4(Y)	27.85/-	-	Online
FDAN [70]	2D CNN	L1 loss	Vimeo-90k	Vimeo-90k-T (Y) Vid4 (Y) UDM10 (Y)	37.75/0.9522 27.88/0.8508 39.91/0.9686	8.97	Online
BasicVSR [13]	Bidirectional RNN	Charbonnier loss	REDS Vimeo-90k Vimeo-90k	REDS4 (RGB) Vimeo-90k-T (Y) Vid4 (Y)	31.42/0.8909 37.18/0.9450 27.24/0.8251	6.3	Offline
IconVSR [13]	Bidirectional RNN	Charbonnier loss	REDS Vimeo-90k Vimeo-90k	REDS4 (RGB) Vimeo-90k-T (Y) Vid4 (Y)	31.67/0.8948 37.47/0.9476 27.39/0.8279	8.7	Offline
GOVSR [71]	Bidirectional RNN	Charbonnier	MM522 MM522	Vid4 (Y) UDM10 (Y)	28.41 / 0.8724 40.14/0.971	1.90	Offline
MSHPFNL [72]	GAN	Weighted (adversarial, perceptual,frame variation)	MM522 MM522 Vimeo-90k	Vid4 (Y) UDM10 (Y) Vimeo-90k-T (Y)	27.70/0.8472 39.59/0.9676 36.75/0.9406	7.77	Online
TTVSR [73]	Transformer	Charbonnier loss	REDS Vimeo-90k Vimeo-90k Vimeo-90k	REDS4 (Y) Vid4 (Y) UDM10 (Y) Vimeo-90k-T (Y)	32.12/0.9021 28.40/0.8643 40.41/0.9712 37.92/0.9526	6.8	Online
BasicVSR++ [74]	Bidirectiona RNN	Charbonnier loss	REDS Vimeo-90k Vimeo-90k	REDS4 (RGB) Vimeo-90k-T (Y) Vid4 (Y)	32.39/0.9069 37.79/0.9500 27.79/0.8400	7.3	Offline
PSRT [75]	Transformer	Charbonnier loss	REDS	REDS4 (Y)	31.32/0.8834	-	Online
R2D2 [40]	RNN	L1 Loss	Vimeo-90k	Vid4 (Y) UDM10 (Y) SPMCS11 (Y)	28.13/0.9244 39.53/0.9670 30.71/0.8920	8.25	Online
R2D2-*lite* [40]	RNN	L1 Loss	Vimeo-90k	Vid4 (Y) UDM10 (Y) SPMCS11 (Y)	28.05/0.8552 39.38/0.9661 29.91/0.8758	6.85	Online

Y and RGB indicate the channel of HR output used for computing quality. In the case of models with different input degradation, as reported in table i, their corresponding bicubic degradation-related results are reported. Size represents model parameters in millions. The application represents the applicability of the model based on input feed.

due to their ability to learn hierarchical features from spatial data. In the context of VSR, 2D CNNs treat each frame independently, essentially performing image super-resolution on every frame by making use of LR frames available in a sliding window. While computationally efficient, this approach neglects the temporal correlations between consecutive frames, which are crucial for video. The convolution filters in 2D CNNs extract spatial features from each frame or fusion of frames and apply activation functions such as ReLU, sigmoid or tanh to introduce non-linearity. These features are then refined and upsampled to generate super-resolved outputs. 2D CNN-based VSR models struggle to handle videos with substantial motion or dynamic scenes due to a lack of temporal information and have become uncommon.

- *3-Dimensional Convolutional Neural Networks (3D CNNs)* extend the concept of 2D CNNs to incorporate temporal information by operating on spatiotemporal data cubes. By applying filters across three dimensions: width, height, and time jointly, these models process both spatial and temporal dimensions producing 3D feature maps and effectively capturing motion patterns and temporal dependencies within a given video segment (usually a sliding window) [41], [42], [51], [66]. This makes 3D CNNs well-suited for tasks involving dynamic scenes and motion-based artifacts. However, the use of 3D convolutions increases the model's computational complexity and memory requirements, making it challenging to scale to longer video sequences or low-resource devices. Moreover, the limited temporal context confined to a fixed temporal radius in a sliding window does not allow these models to exploit the temporal correlation across the video frame series fully.
- *Generative Adversarial Networks (GANs)* have shown impressive results in various image synthesis tasks, including single-image super-resolution [25]. In the context of VSR, GAN-based models consist of a generator network that produces high-resolution video frames and a discriminator network that distinguishes between real high-resolution frames and generated synthetic ones [72], [84]. Adversarial training encourages the generator network to produce visually realistic results. However, training GAN-based VSR models can be challenging due to the inherent instability of GAN training. Training GANs often require extensive hyperparameter tuning and a large quantity of training data to achieve high-quality results. Additionally, GAN-based models may suffer from mode collapse, where they fail to capture the full diversity of the training data distribution and instead generate a limited set of outputs, often repetitive or unrepresentative of the entire dataset. It occurs when the model ignores or collapses multiple modes (distinct patterns or clusters) in the data and focuses on generating only a few dominant modes [85]. GAN-based models' ability to capture temporal information is also limited, similar to that of 2D CNNs, as GAN generators apply 2D convolution operations to extract features from frames in a sliding window, enabling only spatial features capture [72], [84].
- *Encoder-Decoder architectures* have been successful in various image processing tasks, and their application to VSR is likewise promising. These networks consist of an encoder that compresses the low-resolution video frames into a latent representation and a decoder that reconstructs the high-resolution output. Although encoder-decoder networks are known to be used for sequential modelling with RNNs [86], or their variants such as LSTM [87] or gated recurrent unit (GRU) [88] in other domains, in VSR, encoder-decoder networks are commonly been used with sliding windows of fixed length, employing 2D convolution layers for feature extraction and fusion enabling them to learn spatial information only [16], [65]. Thus, encoder-decoder networks struggle to fully capture long-range temporal dependencies, limiting their performance in videos with complex temporal patterns.
- *Transformers* have revolutionised natural language processing tasks and have recently attracted attention in computer vision applications. In the context of VSR, transformers can effectively model temporal relationships between frames in a sliding window by employing self-attention mechanisms [3]. This allows them to weigh the importance of different frames dynamically, capturing temporal variations and correlations within a sliding window. Although transformer use is limited in VSR, they have shown promising results in handling complex temporal patterns and achieving state-of-the-art performance in various other video-related tasks such as object detection, recognition, captioning, segmentation and anomaly detection [89]. However, the computational complexity of transformers is greater than traditional CNN-based architectures, which may hinder their real-time deployment on resource-constrained devices.

2) *Sequential:*

- *Unidirectional RNNs*, using LSTM, GRU or residual blocks, process video frames sequentially from past to future. They have the ability to model temporal dependencies and retain long-term information through recurrent connections. However, their main limitation is the restricted context they consider since they only have access to past frames during training. As a result, unidirectional RNN-based VSR models may struggle to capture complex temporal patterns in videos where bidirectional interactions between frames are prevalent. Moreover, the amount and relevance of the temporal context available for earlier timestamps in a video sequence are inadequate, often causing subpar performance for the initial frames and a gradual improvement over time [39]. Nevertheless, RNNs have proven to be one of the more effective architectures in modelling the VSR task sequentially. Recent advances over vanilla RNNs tend to address the vanishing gradient issue and information availability for earlier frames by using hybrid input feed and alignments [40], [71]. Further integrating learning techniques such as LSTM residual blocks [90], [91], latent representation [63], feature attention [45] and multi-stream refinement [8], [40] have resulted in improved RNN performance for VSR, overcoming earlier challenges and performing competitively with significantly reduced computation and number of frames.
- *Bidirectional RNNs* aim to overcome the limitations of unidirectional RNNs by processing video frames in both forward and backward temporal directions simultaneously. By doing so, Bidirectional RNNs can capture information from past and future frames, providing a more comprehensive context for making super-resolution predictions. Bidirectional RNNs are well-suited for tasks

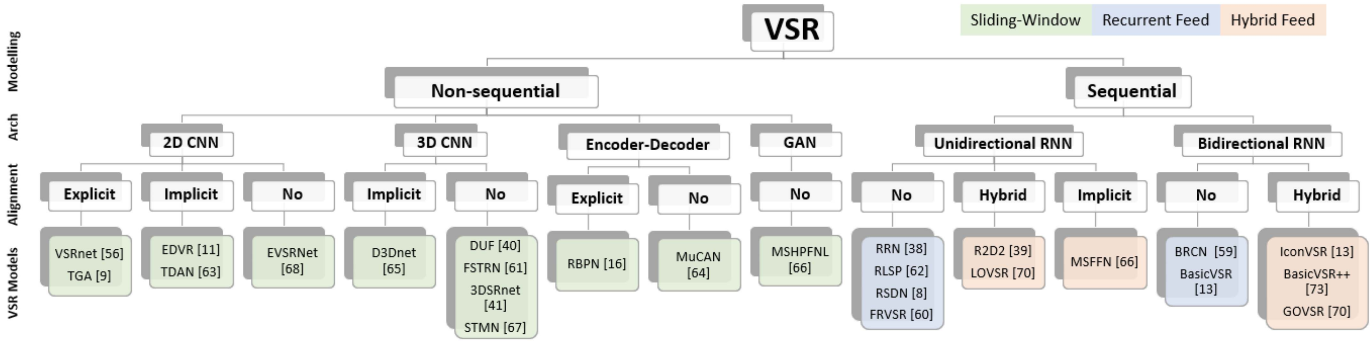

Fig. 4. Highlight of VSR models from the literature based on discussions in Sections III and IV.

involving bidirectional motion or events where future information significantly influences the current frame's super-resolution. However, bidirectional RNNs introduce higher computational costs compared to unidirectional RNNs and need entire video frames to be available during inference, making this approach unfit for several applications related to real-time and online VSR applications.

3) Network Selection: For computational efficiency and real-time applications, 2D CNNs and Encoder-Decoder networks are preferred over 3D CNNs and Transformers. However, when capturing long-range dependencies and handling complex temporal patterns is crucial to the VSR task, and where efficiency is a priority, unidirectional RNNs are more suitable choices. GAN-based VSR models may suffer from training instability and therefore require a larger training dataset, while traditional CNN-based and RNN-based architectures offer more stable training with the potential to perform well using smaller training datasets. Additionally, 3D CNNs and Transformers have greater memory requirements, making 2D CNNs and Encoder-Decoder networks preferable for memory-limited scenarios. Applications with no real-time and online constraints can leverage the full potential of temporal modelling from bidirectional RNNs in offline settings.

The taxonomy presented in Fig. 3 in combination with the learning elements discussed in Section IV and presented in Table II have been combined and summarised in Fig. 4 which provides the merged synopsis of the learning-based modelling technique used by VSR models in literature along with their respective network architecture, alignment technique and input feed mechanism. This helps establish an association between the component methodologies discussed in Section III in relation to the deep learning techniques discussed in Section IV. It is evident from Fig. 4 that non-sequential modelling remains the most common approach to VSR solution development in the literature. It is also evident that sequential modelling using RNNs is becoming popular with no use of explicit or implicit alignment techniques except for hybrid alignment. No alignment technique also appears common for models deploying spatiotemporal feature extraction capabilities such as 3D CNNs and RNNs. Similarly, the sliding window remains the only used approach for input feed for non-sequential models, which is incorporated only by hybrid alignment models in the case of the sequential approach. This helps summarise that the applicability of component methodology is heavily driven and dependent on the learning techniques used.

B. Datasets

Benchmark datasets play a crucial role in the training and testing of deep learning VSR models. As reported in Table II, Vimeo90k [92] is the most commonly used dataset for training because it is the largest dataset in the VSR domain and consists of diverse real-world scenes. The details of widely used datasets are given in Table III. Vid4 [93] is the most common dataset used for test purposes, although it does not contain diverse scenes nor large inter-frame motion [16]. YUV25[1] and CDVL[2] have been used in earlier works for training and evaluation purposes. In order to train and test VSR models that account for video sequences with large inter-frame motion, the most challenging property of video for VSR models, the REDS [94] dataset has been used since 2019. Other datasets, such as Myanmar,[3] Ultra-Video Group[4] and SPMCS [59], used by various published works for test purposes, are also listed and described in Table III.

C. Loss Functions

There are various loss functions used to train VSR models in order to optimise learning. These functions can be broadly categorised into two groups: *pixel-based* and *perception-based.*

1) Pixel-Based: Pixel-based objectives aim to minimise the discrepancy between the generated HR frame and the corresponding ground truth (GT) frame. These objectives measure the pixel-wise differences between the generated HR and GT frames. Several commonly used reconstruction-based loss functions in VSR include:

- *Mean Squared Error (MSE)*, also known as *L2* loss, measures the average of the squared differences between the generated HR frame and the corresponding GT frame. MSE is a commonly used loss function in various image and video processing tasks. Minimising MSE encourages VSR models to generate HR frames that closely match

[1] YUV25 - https://media.xiph.org/video/derf/
[2] CDVL - https://www.cdvl.org/
[3] Myanmar- https://www.harmonicinc.com//insights/blog/4k-in-context/
[4] Ultra-Video Group Test Sequences - http://ultravideo.cs.tut.fi/

TABLE III
BENCHMARK DATASETS COMMONLY USED FOR VIDEO SUPER-RESOLUTION IN THE LITERATURE

Dataset	Resolution	Description
YUV25[1]	386×288p	25 videos in the uncompressed YUV4MPEG format.
Myanmar[3]	3840×2160p	59 scenes, uncompressed 4K resolution.
CDVL[2]	1920 × 1080	100 RGB videos from The Consumer Digital Video Library.
Vid4 [93]	Calendar-720x576p, Walk–720x480p, Foliage-720x480p, City–704x576p	The sequences in this dataset have visual artifacts, very little inter-frame variation, and limited motion. Most notably, the dataset consists of only four video sequences – City, Walk, Calendar and Foliage.
Vimeo90K [92]	448x256p	A septuplet dataset consisting of 91,701 7-frame sequences with a fixed resolution 448x256, extracted from 39K selected video clips from the Vimeo-90K website. This dataset is designed for video de-noising, de-blocking, and super-resolution. Widely used for training as it contains a large number of clips with varied motions.
Ultra-Video Group[4]	1920×1080p	7 HD videos of approx. 5 sec in length
SPMCS [59]	540×960p	975 sequences from commercial videos shot with high-end cameras and containing both natural-world and urban scenes with rich detail. Each sequence contains 31 frames. 945 sequences are randomly chosen as training data, and the rest 30 sequences are for validation and testing.
REDS [94]	1280x720	240 video sequences with 100 frames in each, widely used for training and testing since 2019. Primarily includes clips with large inter-frame motions.

the ground truth in terms of pixel values. However, MSE tends to emphasise large errors, which may lead to over-smoothed results, and MSE can produce blurry outputs. While MSE provides a straightforward and differentiable objective, optimising loss solely based on MSE can produce visually unsatisfactory results. It can, for instance, result in overly smooth outputs that lack fine detail and texture. MSE can be mathematically represented as:

$$L_{MSE} = \frac{1}{N}\sum_{i=0}^{N-1}(GT_i - HR_i)^2 \tag{2}$$

where GT_i represents the ground truth frame pixel at position i, HR_i is the generated HR frame pixel at position i, and N is the total number of pixels.

- *Mean Absolute Error (MAE)*, also known as *L1* loss, calculates the average of the absolute differences between the generated HR frame and the GT frame. Unlike MSE, MAE is less sensitive to outliers, making it more robust to extreme errors. By minimising MAE, the model is encouraged to produce HR frames that have accurate pixel-wise correspondence to the ground truth. MAE loss helps reduce the impact of outliers, which can be beneficial for handling noisy or corrupt data. MAE often leads to sharper and more detailed outputs compared to MSE loss. However, MAE may still produce over-smoothed results. MAE is expressed as:

$$L_{MAE} = \frac{1}{N}\sum_{i=0}^{N-1}|GT_i - HR_i| \tag{3}$$

- *Smooth-L1 Loss*: Also known as Huber loss, combines the benefits of both *L1* and *L2* losses. Smooth-*L1* Loss behaves in the same way as *L2* loss when the pixel-wise differences are small (less sensitive to outliers) and as *L*1 loss when the pixel-wise differences are large. The transition is controlled by a hyperparameter β. Smooth-*L1* loss is advantageous because it suppresses the impact of outliers while still penalising large errors. Smooth-L1 loss can help strike a balance between accuracy and robustness. It tends to produce visually pleasing results with sharper details and fewer artifacts than MSE or MAE. The mathematical representation is:

$$L_{Smooth-L1} = \begin{cases} \frac{1}{N}\sum_{i=0}^{N-1} 0.5(GT_i - HR_i)^2\beta, \text{if } |GT - HR| < \beta \\ \frac{1}{N}\sum_{i=0}^{N-1} |GT_i - HR_i| - 0.5\beta, \text{otherwise} \end{cases} \tag{4}$$

where β is a hyperparameter that controls the point where the loss transitions between *L1* and *L2* behaviour.

- *Charbonnier Loss:* Charbonnier Loss, also known as Charbonnier penalty (aka pseudo-Huber loss), is a differentiable and smooth approximation of *L2* loss while having similar robustness properties as the *L1* loss. By using the square root, Charbonnier loss reduces the weight of large errors and focuses more on smaller errors. This property makes it less sensitive to outliers and, therefore, more robust. Charbonnier loss can be an effective alternative to MSE and MAE, especially when dealing with noisy data or artifacts in the ground truth. It encourages the model to generate HR frames that preserve important visual details while suppressing the impact of noisy or inconsistent data and is defined as:

$$L_{Charbonnier} = \frac{1}{N}\sum_{i=0}^{N-1}\sqrt{(GT_i - HR_i)^2 + \epsilon^2} \tag{5}$$

where ϵ is a small positive constant.

- *Adversarial Loss:* Adversarial loss employs a discriminator network to distinguish between the generated HR frame and real HR frames. It encourages the generated frames to be indistinguishable from the real HR frames. By training the model in an adversarial setting, it learns to capture finer details and textures in the generated HR frames, leading to more natural-looking outputs. Denoting the discriminator's output (probability of the frame being real) as $D(GT)$, for the ground truth HR frame and $D(HR)$, for the generated HR frame. Adversarial loss can subsequently be formulated as follows:

$$L_{Adversarial} = \frac{1}{N}\sum_{i=0}^{N-1}[\log D(GT_i) + \log(1 - D(HR_i))] \quad (6)$$

In this adversarial loss formulation, the generator tries to maximise $\log(1 - D(HR_i))$, aiming to produce HR frames that are as close as possible to the pixel-wise representation of the discriminator. On the other hand, the discriminator tries to maximise $\log D(GT_i)$ and $\log(1 - D(HR_i))$ to improve its ability to distinguish between real and generated frames.

- *Weighted Loss:* Weighted loss balances the contributions of different loss functions during training. By assigning weights to each loss, the model can focus on optimising specific aspects of the generated HR frames and customise the trade-off between different objectives. For example, a combination of pixel-wise loss (e.g., MSE) and perceptual loss can be used to achieve a balance between reconstruction accuracy and visual quality. Multiple loss functions are combined using weighted coefficients:

$$L_{Weighted} = w_1 L_1 + w_2 L_2 + \ldots + w_n L_n \quad (7)$$

where L_j represents the j^{th} loss function, and w_j is the weight assigned to that loss function. The weight to be assigned is determined based on the relative importance of individual loss terms using manual tuning based on domain expertise or empirical validation of the relative impact of the individual components during training.

2) Perception-Based: Perception-based objectives focus on the perceived quality of the generated HR frame rather than pixel-wise differences. These objectives employ pre-trained networks, such as VGGNet [95] and AlexNet [96], to extract features from the GT and generated HR frames, measuring the similarity between these features. The most commonly used perception objective-based loss function in VSR is perceptual loss. Perceptual loss utilises a pre-trained network to extract high-level features from the GT and generated HR frames. By comparing these features, the model is guided to minimise the difference in high-level representations rather than via pixel-wise error analysis. Perceptual loss aims to improve the visual quality of the generated HR frames by focusing on higher-level visual features. It encourages the model to capture the overall structure and content of the frames, resulting in outputs that are visually appealing to human observers. Perceptual loss is particularly effective in reducing the over-smoothing issue often encountered when relying solely on pixel-wise loss functions. Perceptual loss measures the similarity between these features. Mathematically, it can be expressed as:

$$L_{Perceptual} = \frac{1}{M}\sum_{k=0}^{M-1}(F(GT_k) - F(HR_k))^2 \quad (8)$$

where $F(GT_k)$ and $F(HR_k)$ are the features extracted from the ground truth frame and the generated HR frame, respectively, and M is the total number of features.

D. Advanced Training and Optimisation

To improve the performance and efficiency of VSR models, several advanced training and optimisation techniques are employed with the aim of enhancing the learning process and the generalisation capabilities of the VSR models. Some commonly used techniques in VSR include:

1) Transfer Learning: Transfer learning is a technique that leverages pre-trained model weights from other domains and adapts them to the VSR task. By utilising the knowledge learned from related tasks or domains, transfer learning allows VSR models to benefit from the representational power of pre-trained models and enhance their performance with even limited training data. Transfer learning has been used for several purposes in the VSR literature. Initially, it was used to transfer the learning of a single image super-resolution model to a video super-resolution model as a base initialisation of the learning process [57]. With proven advances, transfer learning has also been used to fine-tune deep learning-based optical flow estimation methods such as SpyNet [76], in the particular context of video super-resolution task [13], [40]. More recently, transfer learning has been used to adapt the super-resolution task from conventional 2D videos to the new domain of 360° video super-resolution [45], where the availability of the training dataset is limited. In each instance, transfer learning has proven to be an effective tool in extending the learning and performance ability of VSR models.

2) Escaping Local Minima: Deep learning VSR models are typically trained using gradient descent optimisation, which can sometimes get stuck in local minima, leading to suboptimal solutions. To address this issue, various techniques have been proposed to escape local minima and find better global minima. One such technique is to use a well-known variant of gradient descent called stochastic gradient descent (SGD). SGD introduces noise in the gradients and helps better explore the optimisation landscape. Another technique is to use a variant of SGD called Adam optimisation [97], which adapts the learning rate for each parameter and helps prevent getting stuck in suboptimal local minima. The Adam optimiser remains the most commonly used optimisation technique in recent VSR literature. Recently, metaheuristic optimisation using learning rate reset has also proven to help expand VSR model learning ability despite premature training saturation [40] while still improving inference results confirming no over-fitting. These approaches are guiding new directions in VSR research with an emphasis on advanced training and optimisation efforts.

3) Knowledge Distillation: Knowledge distillation is a technique that allows the transfer of knowledge learned by a complex and larger model (the teacher) to a simpler and smaller model (the student). In the context of VSR, knowledge distillation was first explored by Xiao et al. [98] where the complex EDVR [11] model was used to train a shallower student model for space and time distillation. Space distillation was done to train the student model to produce the teacher-like attention map, while time distillation was performed to help the student model learn weights for Convolution LSTM memory units from the teacher network. Adopting this technique, it was proven that multiple existing shallower networks were able to produce better results compared to conventional training. Several other works have since proven the effectiveness of knowledge distillation in VSR for the design of lightweight real-time VSR models [99], to compress the model size of recurrent network [10] and to optimise the size of MEMC-based super-resolution model [100].

4) Model Compression: Model compression techniques aim to reduce the computational complexity and memory requirements of VSR models while preserving their performance. Techniques such as pruning, quantisation, and low-rank approximation can be applied to compress the model size and make it more suitable for deployment on resource-constrained devices or in real-time applications. Although earlier discussed, knowledge distillation techniques have also been used to train a compressed VSR model, the discussion here is focused on the methodology of compressing a trained network. In order to prune redundant filters, a recent work [101] proposed structured pruning schemes for residual blocks, recurrent networks, and upsampling networks. It was therein proven the application of pruning on BasicVSR [13] resulted in up to ×4 optimisation of efficiency while maintaining competitive qualitative and quantitative results. Agrahari Baniya et al. [40] also applied the pruning technique to optimise the inference efficiency by removing the explicit alignment component from a trained model and instead retaining the alignment knowledge in convolution layers. This approach significantly improves model efficiency during inference enabling the lighter version to be used in an online application context with minimal performance degradation.

Real-time resource-constrained applications are becoming a normative use case for VSR with the forecast growth of handheld and edge devices. Advanced training and optimisation techniques are therefore the key enablers of VSR model development for these requirements [102].

V. Evaluation

A. Quality

Quality evaluation in video super-resolution involves assessing the performance of the generated high-resolution (HR) video frames in reference to the ground truth (GT) frames. The assessment of quality is either subjective or objective. Predominantly, objective assessment methods based on pixel comparisons are used in the VSR domain. While the objective metrics help quantify the quality of produced HR frames and enable an analysis of the impact of Quality of Service (QoS), they often do not align with human perception of VSR quality. Most video-based multimedia is consumed through human perception, and thus there is a growing interest in evaluating VSR model performance in terms of perceptual quality.

1) Peak Signal-to-Noise Ratio (PSNR): PSNR is a widely used objective quality assessment metric in image and video processing tasks. PSNR measures the quality of the reconstructed video frame by comparing each co-located pixel with the original GT frame. PSNR is calculated as the ratio of the maximum possible pixel value to the Mean Squared Error (MSE) between the HR and GT frames. Higher PSNR values indicate better reconstruction quality since its value signifies less distortion and a closer resemblance to the GT. However, PSNR has limitations in capturing perceptual differences and may not always correlate well with human visual perception. PSNR is calculated as:

$$\text{PSNR}_{(GT,HR)} = 10 \log_{10} \left(\frac{MAX^2}{MSE_{GT,HR}} \right) \tag{9}$$

where MAX is the maximum possible pixel value of the image (usually 255 for an 8-bit frame), and $MSE_{GT,HR}$ is the Mean Squared Error between the GT and the reconstructed HR frames computed using (2).

2) Structural Similarity Index Measure (SSIM): SSIM is another objective image quality metric that evaluates the structural similarity between the HR and GT videos by considering luminance, contrast, and structural information. The SSIM index ranges between -1 and 1, with 1 indicating perfect similarity. SSIM accounts for both global and local spatial structural information and is known to correlate better with human perception compared to PSNR. SSIM can be computed as:

$$\text{SSIM}_{(GT,HR)} = \frac{(2\mu_{GT}\mu_{HR} + c_1)(2\sigma_{GT,HR} + c_2)}{(\mu_{GT}^2 + \mu_{HR}^2 + c_1)(\sigma_{GT}^2 + \sigma_{HR}^2 + c_2)} \tag{10}$$

Where μ_{GT} and μ_{HR} are the means of the image GT and HR, σ_{GT}^2 and σ_{HR}^2 are the variances of the image GT and HR, and $\sigma_{GT,HR}$ is the covariance of the image GT and HR. The constants c_1 and c_2 are used to stabilise the division with a weak denominator.

3) Natural Image Quality Evaluator (NIQE): NIQE measures the naturalness of a frame where no reference is provided [103]. The degree of naturalness is computed by analysing image statistics such as luminance, contrast and edge information. Higher NIQE scores indicate higher perceived naturalness in the frame and are useful in assessing the perceived quality of super-resolved frames, as it considers characteristics important for human visual perception beyond traditional pixel-based metrics like PSNR and SSIM.

4) Learned Perceptual Image Patch Similarity (LPIPS): is a perceptual metric that measures the perceptual distance between two frames using a neural network like VGG16. LPIPS quantifies the visual dissimilarity between the HR and GT frames, taking into account high-level features extracted from the VGG model, which has learnt feature extraction from a large database of images. LPIPS is designed to capture human perceptual judgments and is more in line with human visual perception compared to PSNR and SSIM.

$$\text{LPIPS}_{(GT,HR)} = \|VGG_\theta(GT) - VGG_\theta(HR)\|_2 \tag{11}$$

where VGG_θ is the neural network, GT and HR are the frames being compared, and the norm is the $L2$-norm. Unlike the traditional quality metrics such as PSNR and SSIM, LPIPS is a trained metric using a neural network and has only recently started being used for evaluation. It is still not a common assessment metric in the VSR literature; however, it is commonly used to evaluate models in competitions like NTIRE [104].

5) Temporal Consistency: Although not commonly used, it is an important perceptual aspect in VSR to evaluate the temporal coherence between consecutive frames in the generated HR video sequence. Temporal consistency metrics such as flow warping error [105] assess the maintenance of continuity and smoothness over time, ensuring that there are no noticeable artifacts or inconsistencies between frames. High temporal consistency indicates that the VSR model effectively preserves the motion dynamics and temporal details present in the GT, leading to visually pleasing results.

B. *Efficiency*

Efficiency is a critical factor to consider when evaluating video super-resolution models. Traditionally, the key focus on VSR model evaluation was limited to quality assessment. However, with the growing computational demand of neural networks and constraints around resource availability, efficiency evaluation during inference has become a standard evaluation method. This section explores key efficiency metrics, including model size, inference time, frames per second (FPS), and floating-point operations (FLOPs).

1) Model Size: The model size denotes the memory requirements for storing parameters and architecture of a video super-resolution model. Model size influences the feasibility of deploying the model on resource-constrained devices or in scenarios where storage capacity is limited. Smaller model sizes are advantageous, as they enable efficient utilisation of storage resources. The size of a deep learning video super-resolution model can be computed by considering the number of parameters it contains. Denoting the total number of layers in the model as P and the number of parameters in each layer as L_l, where l represents the layer index. The total number of parameters in the model, T, can then be calculated by summing the parameters across all layers:

$$T = \sum_{l=0}^{P-1} L_l \tag{12}$$

Each layer may have different types of parameters, such as weights, biases, or other learnable parameters specific to the layered architecture. Therefore, L_l can vary depending on the layer type and its configuration. For example, in a fully connected (dense) layer, L_l would be the product of the number of input units, n_{in}, and the number of output units, n_{out}, along with any additional parameters such as biases:

$$L_l = n_{\text{in}} \times n_{\text{out}} + n_{\text{out}} \tag{13}$$

Similarly, in a convolutional layer, the number of parameters depends on the filter size, the number of input channels, the number of output channels, and any biases:

$$\begin{aligned} L_l = {} & (\text{filter\_width} \times \text{filter\_height} \times \text{input\_channels} + 1) \\ & \times \text{output\_channels} \end{aligned} \tag{14}$$

By summing the parameters across all layers, as shown in (12), the total size of the deep learning video super-resolution model is determined.

2) Inference Time: Inference time (t_{inf}) represents the duration taken by a model to process a single video frame and generate the corresponding super-resolved output. Minimising inference time is important, particularly for real-time applications, where prompt processing and smooth playback are essential. The inference time of a deep learning video super-resolution model can be computed by considering the number of frames, the model architecture, and the hardware specifications. Denoting the total number of frames in the video as F, and the time taken to process one frame as t_{frame}. Additionally, assume that the model processes each frame independently without any temporal dependencies. The total inference time, T_{total}, can then be calculated by multiplying the number of frames by the time taken to process one frame:

$$t_{\text{total}} = F \times t_{\text{frame}} \tag{15}$$

The time taken to process one frame, t_{frame}, depends on various factors such as the model architecture, the input size of each frame, and the hardware specifications. The time taken can be further decomposed into the time taken for computation, memory access, and any additional overhead:

$$t_{\text{frame}} = t_{\text{computation}} + t_{\text{memory}} + t_{\text{overhead}} \tag{16}$$

The computation time, $t_{\text{computation}}$, represents the time taken by the model to perform the necessary computations for each frame. This depends on the complexity of the model architecture and the number of operations required. The memory access time, t_{memory}, accounts for the time taken to load input frames, intermediate results, and store output frames in memory. This depends on the memory bandwidth and access patterns. The overhead time, t_{overhead}, includes any additional time required for data preprocessing, postprocessing, or any other operations specific to the model or the implementation. By summing the computation, memory access, and overhead times as shown in (16), the time taken to process one frame can be determined. Multiplying it with the number of frames, as shown in (15), gives the total inference time of the video super-resolution model.

3) Frames Per Second (FPS): The Frames per Second (FPS) metric measures the number of video frames that a model can process per second. Higher FPS values indicate faster processing capability and improved real-time performance. Models with high FPS can effectively handle videos with higher frame rates, ensuring seamless and continuous output without compromising quality. FPS can be computed as the inverse of the inference time per frame. Given that the inference time for a frame is in seconds, FPS is:

$$FPS = \lfloor 1/t_{\text{frame}} \rfloor \tag{17}$$

where $\lfloor\ \rfloor$ represents the rounded value for whole number.

4) Floating Point Operations (FLOPs): Floating Point Operations (FLOPs) provide an estimation of the computational complexity of a video super-resolution model. FLOPs quantify the number of floating-point arithmetic operations required for each inference. Evaluating FLOPs is essential, as lower FLOP counts generally indicate more computationally efficient models. Such models are advantageous for deployment on devices with limited computational power or in scenarios where energy consumption is a concern. The computation of FLOPs can be estimated using equations based on the model architecture and the number of operations performed by each layer. These equations depend on the specific architecture and operations involved and can be derived accordingly.

VI. Applications

A. Spatial Sciences and Geography

Video super-resolution techniques have proven to be invaluable in the field of spatial sciences and geography. By enhancing the spatial resolution of video content, finer details and identifying patterns can be extracted, enabling accurate analysis and understanding of various geographical phenomena. An application of video super-resolution in this domain is for satellite imagery that captures large amounts of video data from space, but the resolution is often limited due to complex imaging conditions and unknown degradation process [106], [107]. VSR can enhance the resolution of satellite videos allowing for more precise monitoring of land cover changes, urban growth, and environmental phenomena such as deforestation or glacier melting [108], [109]. Additionally, STVSR can also be used to obtain enhanced motion dynamics for extreme events observation while further improving spatial super-resolution [110]. However, challenges persist in variation of scale across the satellite imagery and scarce motion in long-time series satellite video frames. Additionally, video super-resolution techniques find application in geographic surveillance systems. Surveillance cameras installed in urban or natural environments often capture low-resolution video footage, which can hinder the identification of important details. Super-resolution algorithms can enhance the resolution of these videos, enabling more accurate object detection, tracking, and recognition. This is particularly useful in scenarios such as crowd monitoring or traffic analysis.

B. Streaming Entertainment

Video streaming in the entertainment industry has greatly benefited from video super-resolution techniques, enhancing the viewing experience for audiences. With the increasing demand for high-quality content on Video streaming platforms such as Netflix, Amazon Prime, and YouTube, video super-resolution plays a vital role in upscaling lower-resolution videos to meet the expectations of its viewers.Streaming services can enhance the resolution of videos in real-time or during post-production [1] allowing user content to have improved visual quality providing an improved viewing experience [111]. Super-resolution techniques also help preserve the integrity of older or archived content by upscaling it to modern display standards. Similarly, specialised super-resolution networks, such as those tailored for face super-resolution employing purpose-built network architectures [112] can further assist in improving the perceived QoE of the videos streamed.

C. Extended Reality

Extended reality (XR), which encompasses virtual reality (VR), augmented reality (AR), and mixed reality (MR), relies heavily on high-resolution video content for creating realistic and immersive experiences. Video super-resolution techniques play a crucial role in enhancing the visual quality of XR content. In VR applications, super-resolution can be used to improve the resolution and clarity of 360° videos, creating more immersive virtual environments. Higher-resolution videos reduce pixelation and blurriness, providing users with a more detailed and realistic VR experience [45], [47]. In AR and MR applications, video super-resolution can enhance the quality of real-time video feeds from cameras, allowing virtual objects or information to be seamlessly integrated into the user's view [113]. By upscaling the video feed, super-resolution ensures that the virtual elements blend more smoothly with the real environment, enhancing the overall visual fidelity and user experience.

D. Agriculture

Video super-resolution has significant applications in the field of agriculture, enabling more detailed and accurate analysis of crop health, vegetation patterns, and agricultural processes. By enhancing the resolution of aerial or ground-based videos, researchers and farmers can gain valuable insights into crop management and precision agriculture. In aerial imaging, drones equipped with cameras capture videos of agricultural fields. However, due to altitude, distance, and limitations of drone cameras, the captured footage may have a lower resolution. Video super-resolution can help overcome this limitation by enhancing the resolution, allowing for more precise monitoring of crop growth, disease detection, and assessment of irrigation needs. Similarly, in ground-based imaging, video super-resolution can enhance the resolution of videos captured by fixed or moving cameras in the field [114]. This aids in monitoring plant growth, detecting anomalies, and optimising farming practices. The higher resolution enables the identification of small-scale variations in crop health, pest infestations, or nutrient deficiencies, facilitating timely interventions for improved yield and sustainability.

E. Health

Video super-resolution techniques find many applications in the healthcare industry, enabling enhanced visualization and analysis of medical imaging videos [115]. Medical professionals often rely on high-resolution videos for accurate diagnosis, treatment planning, and surgical interventions. In medical imaging, such as endoscopy or laparoscopy, video super-resolution can improve the visibility of fine structures, allowing physicians to detect abnormalities or lesions that may have been difficult to observe in lower-resolution video signals. This assists in the early detection of diseases, precise surgical guidance, and minimally invasive procedures. Furthermore, in telemedicine and remote patient monitoring, video super-resolution plays a vital role in transmitting high-quality video feeds over limited bandwidth connections. By enhancing the resolution of real-time video streams, healthcare providers can assess patients' conditions more accurately, enabling remote diagnosis, consultation, and continuous monitoring.

F. Transport and Road

Video super-resolution techniques have practical applications in the domain of transport and road infrastructure, contributing to improved safety, traffic management, and surveillance systems. In traffic monitoring, surveillance cameras installed on roads capture video footage of vehicles and pedestrians. However, these videos often suffer from low resolution, making it challenging to identify license plates, read signage, or detect fine details for incident analysis [116]. In autonomous vehicles and advanced driver-assistance systems, (ADAS) [117], high-resolution video inputs are crucial for accurate object detection, lane tracking, and collision avoidance. Super-resolution techniques can enhance the resolution of camera feeds, providing a clearer view of the surroundings and improving the performance and safety of autonomous systems. Additionally, video super-resolution is valuable in infrastructure monitoring, where cameras are deployed to assess the condition of roads, bridges, and tunnels. Enhanced resolution videos help identify structural defects, cracks, or signs of deterioration, allowing for timely repairs and proactive maintenance and safety strategies.

VII. Challenges and Trends

A. Online VSR

A growing trend in the current VSR literature, as seen in Tables I and II, is to model the temporal correlation across the time domain effectively. To do so, non-sequential models tend to extend the size of the sliding window to increase temporal context from a wider radius [16], and sequential models tend to use bidirectional recurrent models which expect all the video frames to be available during inference [13]. Both approaches are limited in their ability for online usage such as streaming, video conference, etc., where limited frames are available for each timestamp. One ideal solution is to use unidirectional RNNs, which have proven to be effective while using single frame input per timestamp. However, the lack of information for earlier frames results in compromised performance for the earlier frames in the video series [39].

B. Real VSR

While deep learning VSR models have proven to be effective in restoring high-frequency details from low-resolution video frames downgraded synthetically, their extension to real VSR also known as blind VSR, where the degradation is unknown, remains challenging. Real VSR or blind VSR should encompass diverse degradations, and a single dataset may not represent this diversity well. To address these challenges, modifications involving a cleaning task as a pre-processing step [55], latent feature transformation [53], and diverse kernels combined into VSR for input adoption [6], have been proposed in the literature. Additionally, a decomposition-based learning scheme has been proposed, where LR-HR videos are converted into YUV colour space and the luminance channel is decomposed into a Laplacian pyramid, followed by applying different loss functions to different components, resulting in VSR models with improved performance under real-world settings [118]. Approaches to synthetically create larger datasets with randomised degradation in generating LR video sequences have also been used to train end-to-end VSR models [119]. Unfortunately, all methodologies in deep learning VSR attempt to build a single model solution to generalise the diverse degradations in blind VSR tasks. Alternative methodologies are for diverse degradations in real video inputs without attempting to train a single model to super-resolve all diverse blind video input types.

C. Resource Constrained VSR

Resource-constrained VSR is a significant challenge when deploying super-resolution models on devices with limited computational resources. Real-time applications and on-device processing require efficient algorithms that can produce high-quality results without overwhelming the limited hardware [102]. To address resource constraints, lightweight neural network architectures, those that strike a balance between model complexity and performance, such as MobileNet [120] or EfficientNet [121], have been proposed. These models can significantly reduce the number of parameters and computational costs while maintaining reasonable super-resolution quality. Additionally, techniques such as knowledge distillation [122] and network pruning [40] have been employed to reduce the model size further and improve inference speed without significant loss in VSR accuracy. However, the need for higher resolution and faster inference is growing as technology and consumer behaviours rapidly evolve, resulting need to further optimise VSR computational efficiency.

D. Large and Scalable Upscaling

Large upscaling in VSR involves increasing the resolution of low-resolution videos by a significant factor, typically beyond ×4. Handling large upscaling factors presents unique challenges, as the models need to generate high-frequency details that are significantly missing in the low-resolution input.VSRnet [57] demonstrated the performance degradation as the scaling factor increased from ×2 to ×3 to ×4. Considering ×4 super-resolution as the most representative task, many of the recent VSR models only train and evaluate the model for ×4 super-resolution. The ability of current VSR models to super-resolve beyond that factor and their corresponding solutions remain mostly unexplored in the literature. On the other hand, scalable upscaling expects VSR models to generate a wide range of upscaling factors effectively. For real-world applications, different videos may require various upscaling factors based on their original resolution and the desired output resolution. Fixed-factor VSR models may not be optimal in such scenarios, as they may either overspend computational resources on small upscaling factors or fail to produce satisfactory results for larger factors. A prominent trend in scalable upscaling is the use of progressive upscaling techniques, where the models generate intermediate resolutions before reaching the final high-resolution output. Progressive upscaling allows the model to adapt to different upscaling factors, producing improved results across a wide range of resolutions. For instance, models like RBPN [16] have demonstrated success in scalable upscaling by progressively increasing the spatial dimensions of the generated frames, thereby catering to different upscaling requirements efficiently. In real-world uses, VSR models need to attain enhanced adaptability as well as performance with varying scaling factors.

E. Multi-Objective VSR

Multi-objective VSR refers to the task of simultaneously addressing multiple objectives or criteria in the video super-resolution process. These objectives may include enhancing resolution, improving visual quality, reducing artifacts, and preserving important visual details. Traditionally, VSR models have focused on a single objective, such as maximising a pixel-based accuracy, such as peak signal-to-noise ratio (PSNR) or structural similarity (SSIM) index, to measure the similarity between the super-resolved and ground truth frames. In multi-objective VSR, the goal is to strike a balance between different objectives, considering that they may sometimes conflict with each other. For example, increasing the resolution might introduce artifacts or result in over-smoothed textures, negatively affecting visual quality. To address this challenge, recent research has focused on developing VSR models that can handle multiple objectives simultaneously. One common approach in multi-objective VSR is to use advanced weighted loss functions that combine different metrics, such as a combination of pixel-based losses and perceptual loss based on deep features extracted from pre-trained neural networks such as VGG or ResNet. These combined loss functions allow the model to be optimised for both perceptual quality and objective measures simultaneously. Another trend in multi-objective VSR is to introduce regularisation terms, or additional components in the loss function, that explicitly encourage specific desirable characteristics. For example, to improve the sharpness of super-resolved frames, an edge-aware regularisation term can be included in the loss function to preserve fine details and avoid over-smoothing. Training multi-objective VSR models can be challenging since it involves finding the appropriate trade-offs between different objectives and fine-tuning model parameters accordingly. Moreover, multi-objective VSR may add complexity to the optimisation process, potentially leading to longer training times, difficult convergence and increased computational resources.

F. Space-Time VSR

Space-time VSR (STVSR) models aim to improve video quality across both spatial and temporal dimensions. Incorporating the increased dimensionality of the temporal interpolation directly into conventional VSR without introducing artifacts or distortions while maintaining computational efficiency is challenging. As such, research in STVSR is directed towards efficiently leveraging spatiotemporal correlations while addressing the challenges of simultaneously interpolating time and space details [123]. Recent advancements in STVSRs utilise deep learning techniques such as spatial-temporal transformers [124], deformable convolution LSTMs [125], [126] and mutual learning through iterative up and downsampling [127] to jointly model spatial and temporal dependencies. Temporal profile along with spatial-temporal fusion has also been used to directly exploit the spatial-temporal correlation in the long-term temporal context without the need for motion compensation [128]. On the other hand, synthesising LR frames using flow maps and blending masks, and reusing upsampled versions of these for coarse estimation of HR intermediate frame followed by refinement using residual learning has also resulted in light-weight STVSR models [129]. Despite these advancements, addressing issues related to computational complexity, scalability to high-resolution videos, and generalisation across diverse video content are among the key challenges in this field.

VIII. Conclusion

A thorough overview and systematic categorisation of each VSR component and technology have been presented in this work contributing to a categorical taxonomy and summarisation of VSR practices with highlights of key methods and their implications across VSR stages. Despite the diverse spread of the tools and technologies used in VSR, the critical review introduced in this paper thoroughly summarises the preeminent choices. Furthermore, this study provides detailed guidelines for the deep learning technologies that are currently used in the literature and unpacks the substance of each technology, ranging from VSR model architecture to model training and evaluation. With additional discussion of the application of VSR models and the challenges and trends in the field, this study aims to foster a guide for VSR research with explainable technology selections and requirement-specific modelling. Overall, this study serves as a comprehensive resource for researchers, practitioners, and stakeholders in the field of VSR, facilitating practice-based decision-making and advancing the state-of-the-art in deep learning-based video super-resolution. There needs to be further analysis into the explainability of the combined impact of these components in relation to the results obtained, and this work is intended to set the stage for that.

References

[1] Y. Zhang et al., “Improving quality of experience by adaptive video streaming with super-resolution,” in *Proc. IEEE INFOCOM -IEEE Conf. Comput. Commun.*, 2020, pp. 1957–1966.

[2] K. Purohit, S. Mandal, and A. Rajagopalan, “Mixed-dense connection networks for image and video super-resolution,” *Neurocomputing*, vol. 398, pp. 360–376, 2020.

[3] W. Sun and Y. Zhang, “Attention-guided dual spatial-temporal non-local network for video super-resolution,” *Neurocomputing*, vol. 406, pp. 24–33, 2020.

[4] X. Du, Y. Zhou, Y. Chen, Y. Zhang, J. Yang, and D. Jin, “Dense-connected residual network for video super-resolution,” in *Proc. IEEE Int. Conf. Multimedia Expo*, 2019, pp. 592–597.

[5] M.-H. Cheng, K.-S. Hwang, J.-H. Jeng, and N.-W. Lin, “Classification-based video super-resolution using artificial neural networks,” *Signal Process.*, vol. 93, no. 9, pp. 2612–2625, 2013.

[6] S. Lee, M. Choi, and K. M. Lee, “DynaVSR: Dynamic adaptive blind video super-resolution,” in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2021, pp. 2092–2101.

[7] P. Krämer, J. Benois-Pineau, and J.-P. Domenger, “Local object-based super-resolution mosaicing from low-resolution video,” *Signal Process.*, vol. 91, no. 8, pp. 1771–1780, 2011.

[8] T. Isobe, X. Jia, S. Gu, S. Li, S. Wang, and Q. Tian, “Video super-resolution with recurrent structure-detail network,” in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 645–660.

[9] T. Isobe et al., “Video super-resolution with temporal group attention,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8005–8014.

[10] G. Liu, X. Wang, D. Zha, L. Wang, and L. Zhao, “Efficient, low-cost, real-time video super-resolution network,” in *Proc. Neural Inf. Process.: 28th Int. Conf.*, 2021, pp. 200–211.

[11] X. Wang, K. C. Chan, K. Yu, C. Dong, and C. C. Loy, “EDVR: Video restoration with enhanced deformable convolutional networks,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019, pp. 1954–1963.

[12] D. Li, Z. Wang, and J. Yang, “Video super-resolution with inverse recurrent net and hybrid local fusion,” *Neurocomputing*, vol. 489, pp. 40–51, 2022.

[13] K. C. Chan, X. Wang, K. Yu, C. Dong, and C. C. Loy, “BasicVSR: The search for essential components in video super-resolution and beyond,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4945–4954.

[14] B. Cohen, V. Avrin, and I. Dinstein, “Polyphase back-projection filtering for resolution enhancement of image sequences,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2000, pp. 2171–2174.

[15] G. H. Costa and J. C. M. Bermudez, “Statistical analysis of the LMS algorithm applied to super-resolution image reconstruction,” *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 2084–2095, May 2007.

[16] M. Haris, G. Shakhnarovich, and N. Ukita, “Recurrent back-projection network for video super-resolution,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3892–3901.

[17] N. Fang and Z. Zhan, “High-resolution optical flow and frame-recurrent network for video super-resolution and deblurring,” *Neurocomputing*, vol. 489, pp. 128–138, 2022.

[18] H. Liu et al., “Video super-resolution based on deep learning: A comprehensive survey,” *Artif. Intell. Rev.*, vol. 55, no. 8, pp. 5981–6035, 2022.

[19] H. Chen et al., “Real-world single image super-resolution: A brief review,” *Inf. Fusion*, vol. 79, pp. 124–145, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253521001792

[20] D. C. Lepcha, B. Goyal, A. Dogra, and V. Goyal, “Image super-resolution: A comprehensive review, recent trends, challenges and applications,” *Inf. Fusion*, vol. 91, pp. 230–260, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253522001762

[21] A. A. Efros and T. K. Leung, “Texture synthesis by non-parametric sampling,” in *Proc. IEEE Seventh Int. Conf. Comput. Vis.*, 1999, pp. 1033–1038.

[22] K. I. Kim and Y. Kwon, “Example-based learning for single-image super-resolution,” in *Proc. 30th DAGM Symp. Pattern Recognit.*, 2008, pp. 456–465.

[23] J. Sun, Z. Xu, and H.-Y. Shum, “Image super-resolution using gradient profile prior,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[24] Y.-W. Tai, S. Liu, M. S. Brown, and S. Lin, “Super resolution using edge prior and single image detail synthesis,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 2400–2407.

[25] Z. Wang, J. Chen, and S. C. Hoi, “Deep learning for image super-resolution: A survey,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3365–3387, Oct. 2021.

[26] J. Su, B. Xu, and H. Yin, “A survey of deep learning approaches to image restoration,” *Neurocomputing*, vol. 487, pp. 46–65, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231222002089

[27] C. Dong, C. C. Loy, K. He, and X. Tang, “Learning a deep convolutional network for image super-resolution,” in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 184–199.

[28] Z. Lu, J. Li, H. Liu, C. Huang, L. Zhang, and T. Zeng, “Transformer for single image super-resolution,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 456–465.

[29] K. Jiang, Z. Wang, P. Yi, and J. Jiang, “Hierarchical dense recursive network for image super-resolution,” *Pattern Recognit.*, vol. 107, 2020, Art. no. 107475.

[30] Y. Mei, Y. Fan, and Y. Zhou, “Image super-resolution with non-local sparse attention,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3516–3525.

[31] X. Zhang, H. Zeng, S. Guo, and L. Zhang, “Efficient long-range attention network for image super-resolution,” in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 649–667.

[32] K. Watanabe, Y. Iwai, T. Haga, and M. Yachida, “A fast algorithm of video super-resolution using dimensionality reduction by DCT and example selection,” in *Proc. 19th Int. Conf. Pattern Recognit.*, 2008, pp. 1–5.

[33] D. Kong, M. Han, W. Xu, H. Tao, and Y. Gong, “Video super-resolution with scene-specific priors,” in *Proc. Brit. Mach. Vis. Conf.*, 2006, pp. 549–558.

[34] C.-C. Hsu, L.-W. Kang, and C.-W. Lin, “Temporally coherent superresolution of textured video via dynamic texture synthesis,” *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 919–931, Mar. 2015.

[35] M. Elad and A. Feuer, “Super-resolution reconstruction of image sequences,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 9, pp. 817–834, Sep. 1999.

[36] C. B. Newland, D. A. Gray, and D. Gibbins, “Modified Kalman filtering for image super-resolution: Experimental convergence results,” in *Proc. Signal Image Process.*, 2007, pp. 53–58.

[37] M. Elad and A. Feuer, “Superresolution restoration of an image sequence: Adaptive filtering approach,” *IEEE Trans. Image Process.*, vol. 8, no. 3, pp. 387–395, Mar. 1999.

[38] R. Liao, X. Tao, R. Li, Z. Ma, and J. Jia, “Video super-resolution via deep draft-ensemble learning,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 531–539.

[39] T. Isobe, F. Zhu, X. Jia, and S. Wang, “Revisiting temporal modeling for video super-resolution,” 2020, *arXiv:2008.05765*.

[40] A. A. Baniya, T.-K. Lee, P. W. Eklund, S. Aryal, and A. Robles-Kelly, “Online video super-resolution using information replenishing unidirectional recurrent model,” *Neurocomputing*, vol. 546, 2023, Art. no. 126355.

[41] Y. Jo, S. W. Oh, J. Kang, and S. J. Kim, “Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3224–3232.

[42] S. Y. Kim, J. Lim, T. Na, and M. Kim, “3DSRnet: Video super-resolution using 3D convolutional neural networks,” 2018, *arXiv:1812.09079*.

[43] F. Li, H. Bai, and Y. Zhao, “Learning a deep dual attention network for video super-resolution,” *IEEE Trans. Image Process.*, vol. 29, pp. 4474–4488, 2020.

[44] V. Fakour-Sevom, E. Guldogan, and J.-K. Kämäräinen, “360 panorama super-resolution using deep convolutional networks,” in *Proc. Int. Conf. Comput. Vis. Theory Appl.*, vol. 4, pp. 159–165, 2018.

[45] A. A. Baniya, T.-K. Lee, P. W. Eklund, and S. Aryal, “Omnidirectional video super-resolution using deep learning,” *IEEE Trans. Multimedia*, vol. 26, pp. 540–554, 2024.

[46] M. Dasari, A. Bhattacharya, S. Vargas, P. Sahu, A. Balasubramanian, and S. R. Das, “Streaming 360-degree videos using super-resolution,” in *Proc. IEEE INFOCOM IEEE Conf. Comput. Commun.*, 2020, pp. 1977–1986.

[47] H. Liu et al., “A single frame and multi-frame joint network for 360-degree panorama video super-resolution,” 2020, *arXiv:2008.10320*.

[48] K. Bhandari, Z. Zong, and Y. Yan, “Revisiting optical flow estimation in 360 videos,” in *Proc. 25th Int. Conf. Pattern Recognit.*, 2021, pp. 8196–8203.

[49] T. Tung, S. Nobuhara, and T. Matsuyama, “Simultaneous super-resolution and 3D video using graph-cuts,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[50] Y. Xie, J. Xiao, T. Tillo, Y. Wei, and Y. Zhao, “3D video super-resolution using fully convolutional neural networks,” in *Proc. IEEE Int. Conf. Multimedia Expo*, 2016, pp. 1–6.

[51] J. Li, W. Gao, and Y. Wu, “High-quality 3D reconstruction with depth super-resolution and completion,” *IEEE Access*, vol. 7, pp. 19370–19381, 2019.

[52] M. Joachimiak, M. M. Hannuksela, and M. Gabbouj, “View synthesis quality mapping for depth-based super resolution on mixed resolution 3D video,” in *Proc. 3DTV-Conf.: True Vis.-Capture, Transmiss. Display 3D Video*, 2014, pp. 1–4.

[53] J. Pan, H. Bai, J. Dong, J. Zhang, and J. Tang, “Deep blind video super-resolution,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 4791–4800.

[54] E. Faramarzi, D. Rajan, F. C. Fernandes, and M. P. Christensen, “Blind super resolution of real-life video sequences,” *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1544–1555, Apr. 2016.

[55] K. C. Chan, S. Zhou, X. Xu, and C. C. Loy, “Investigating tradeoffs in real-world video super-resolution,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 5952–5961.

[56] A. A. Baniya et al., “Frame selection using spatiotemporal dynamics and key features as input pre-processing for video super-resolution models,” *SN Comput. Sci.*, vol. 5, no. 3, p. 323, Mar. 2024.

[57] A. Kappeler, S. Yoo, Q. Dai, and A. K. Katsaggelos, “Video super-resolution with convolutional neural networks,” *IEEE Trans. Comput. Imag.*, vol. 2, no. 2, pp. 109–122, Jun. 2016.

[58] J. Caballero et al., "Real-time video super-resolution with spatio-temporal networks and motion compensation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2848–2857.

[59] X. Tao, H. Gao, R. Liao, J. Wang, and J. Jia, "Detail-revealing deep video super-resolution," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4482–4490.

[60] Y. Huang, W. Wang, and L. Wang, "Video super-resolution via bidirectional recurrent convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 1015–1028, Apr. 2018.

[61] M. S. M. Sajjadi, R. Vemulapalli, and M. Brown, "Frame-recurrent video super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6626–6634.

[62] S. Li, F. He, B. Du, L. Zhang, Y. Xu, and D. Tao, "Fast spatio-temporal residual network for video super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10514–10523.

[63] D. Fuoli, S. Gu, and R. Timofte, "Efficient video super-resolution through recurrent latent space propagation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop*, 2019, pp. 3476–3485.

[64] Y. Tian, Y. Zhang, Y. Fu, and C. Xu, "TDAN: Temporally-deformable alignment network for video super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3357–3366.

[65] W. Li, X. Tao, T. Guo, L. Qi, J. Lu, and J. Jia, "Mucan: Multi-correspondence aggregation network for video super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 335–351.

[66] X. Ying, L. Wang, Y. Wang, W. Sheng, W. An, and Y. Guo, "Deformable 3D convolution for video super-resolution," *IEEE Signal Process. Lett.*, vol. 27, pp. 1500–1504, 2020.

[67] H. Song, W. Xu, D. Liu, B. Liu, Q. Liu, and D. N. Metaxas, "Multi-stage feature fusion network for video super-resolution," *IEEE Trans. Image Process.*, vol. 30, pp. 2923–2934, 2021.

[68] X. Zhu, Z. Li, J. Lou, and Q. Shen, "Video super-resolution based on a spatio-temporal matching network," *Pattern Recognit.*, vol. 110, 2021, Art. no. 107619. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320320304222

[69] S. Liu et al., "EVSRNet: Efficient video super-resolution with neural architecture search," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2021, pp. 2480–2485.

[70] J. Lin, Y. Huang, and L. Wang, "FDAN: Flow-guided deformable alignment network for video super-resolution," 2021, *arXiv:2105.05640*.

[71] P. Yi et al., "Omniscient video super-resolution," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 4409–4418.

[72] P. Yi, Z. Wang, K. Jiang, J. Jiang, T. Lu, and J. Ma, "A progressive fusion generative adversarial network for realistic and consistent video super-resolution," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2264–2280, May 2022.

[73] C. Liu, H. Yang, J. Fu, and X. Qian, "Learning trajectory-aware transformer for video super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 5677–5686.

[74] K. C. Chan, S. Zhou, X. Xu, and C. C. Loy, "BasicVSR++: Improving video super-resolution with enhanced propagation and alignment," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 5962–5971.

[75] S. Shi, J. Gu, L. Xie, X. Wang, Y. Yang, and C. Dong, "Rethinking alignment in video super-resolution transformers," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2022, pp. 36081–36093. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/ea4d65c59073e8faf79222654d25fbe2-Paper-Conference.pdf

[76] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2720–2729.

[77] J. Dai et al., "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 764–773.

[78] K. C. Chan, X. Wang, K. Yu, C. Dong, and C. C. Loy, "Understanding deformable alignment in video super-resolution," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 973–981.

[79] P. Yi, Z. Wang, K. Jiang, J. Jiang, and J. Ma, "Progressive fusion video super-resolution network via exploiting non-local spatio-temporal correlations," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 3106–3115.

[80] Y. Xiao et al., "Local-global temporal difference learning for satellite video super-resolution," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 4, pp. 2789–2802, Apr. 2024.

[81] J. Yu, J. Liu, L. Bo, and T. Mei, "Memory-augmented non-local attention for video super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 17813–17822.

[82] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep Laplacian pyramid networks for fast and accurate super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5835–5843.

[83] W. Shi et al., "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1874–1883. [Online]. Available: https://ieeexplore.ieee.org/document/7780576/

[84] A. Lucas, S. Lopez-Tapia, R. Molina, and A. K. Katsaggelos, "Generative adversarial networks and perceptual losses for video super-resolution," *IEEE Trans. Image Process.*, vol. 28, no. 7, pp. 3312–3327, Jul. 2019.

[85] A. Jabbar, X. Li, and B. Omar, "A survey on generative adversarial networks: Variants, applications, and training," *ACM Comput. Surv.*, vol. 54, no. 8, pp. 1–49, 2021.

[86] K. Cho et al., "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014.

[87] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "Lstm-based encoder-decoder for multi-sensor anomaly detection," 2016, *arXiv:1607.00148*.

[88] X.-B. Jin et al., "Deep-learning forecasting method for electric power load via attention-based encoder-decoder with Bayesian optimization," *Energies*, vol. 14, no. 6, 2021, Art. no. 1596.

[89] K. Han et al., "A survey on vision transformer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 87–110, Jan. 2023.

[90] X. Zhu, Z. Li, X.-Y. Zhang, C. Li, Y. Liu, and Z. Xue, "Residual invertible spatio-temporal network for video super-resolution," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 5981–5988.

[91] Z. Wang et al., "Multi-memory convolutional neural network for video super-resolution," *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2530–2544, May 2019.

[92] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *Int. J. Comput. Vis.*, vol. 127, no. 8, pp. 1106–1125, 2019.

[93] C. Liu and D. Sun, "On Bayesian adaptive video super resolution," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 2, pp. 346–360, Feb. 2014.

[94] S. Nah et al., "NTIRE 2019 challenge on video deblurring and super-resolution: Dataset and study," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019, pp. 1996–2005.

[95] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015.

[96] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[97] D. P. Kingma and J. Ba, "ADAM: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[98] Z. Xiao, X. Fu, J. Huang, Z. Cheng, and Z. Xiong, "Space-time distillation for video super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 2113–2122.

[99] J. Xiao et al., "Online video super-resolution with convolutional kernel bypass grafts," *IEEE Trans. Multimedia*, vol. 25, pp. 8972–8987, 2023.

[100] J. Lee and S.-h. Park, "Knowledge distillation for optical flow-based video super-resolution," *J. Comput. Sci. Eng.*, vol. 17, no. 1, pp. 13–19, 2023.

[101] B. Xia et al., "Structured sparsity learning for efficient video super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 22638–22647.

[102] A. Ignatov et al., "Real-time video super-resolution on smartphones with deep learning, mobile AI 2021 challenge: Report," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 2535–2544.

[103] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a "completely blind" image quality analyzer," *IEEE Signal Process. Lett.*, vol. 20, no. 3, pp. 209–212, Mar. 2013.

[104] Y. Li et al., "NTIRE 2023 challenge on efficient super-resolution: Methods and results," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 1922–1960.

[105] W.-S. Lai, J.-B. Huang, O. Wang, E. Shechtman, E. Yumer, and M.-H. Yang, "Learning blind video temporal consistency," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 170–185.

[106] Y. Xiao, X. Su, Q. Yuan, D. Liu, H. Shen, and L. Zhang, "Satellite video super-resolution via multiscale deformable convolution alignment and temporal grouping projection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5610819.

[107] K. Jiang, Z. Wang, P. Yi, and J. Jiang, “A progressively enhanced network for video satellite imagery superresolution,” *IEEE Signal Process. Lett.*, vol. 25, no. 11, pp. 1630–1634, Nov. 2018.

[108] H. Liu, Y. Gu, T. Wang, and S. Li, “Satellite video super-resolution based on adaptively spatiotemporal neighbors and nonlocal similarity regularization,” *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 12, pp. 8372–8383, Dec. 2020.

[109] Y. Xiao, Q. Yuan, Q. Zhang, and L. Zhang, “Deep blind super-resolution for satellite video,” *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5516316.

[110] Y. Xiao et al., “Space-time super-resolution for satellite video: A joint framework based on multi-scale spatial-temporal transformer,” *Int. J. Appl. Earth Observation Geoinf.*, vol. 108, 2022, Art. no. 102731.

[111] A. Zhang et al., “Video super-resolution and caching—An edge-assisted adaptive video streaming solution,” *IEEE Trans. Broadcast.*, vol. 67, no. 4, pp. 799–812, Dec. 2021.

[112] F. Yu, H. Li, S. Bian, and Y. Tang, “An efficient network design for face video super-resolution,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 1513–1520.

[113] G. Mora-Martín, S. Scholes, A. Ruget, R. Henderson, J. Leach, and I. Gyongy, “Video super-resolution for single-photon LIDAR,” *Opt. Exp.*, vol. 31, no. 5, pp. 7060–7072, 2023.

[114] A. A. Baniya, T.-K. Lee, P. W. Eklund, and S. Aryal, “Current state, data requirements and generative AI solution for learning-based computer vision in horticulture,” 2023, doi: 10.20944/preprints202306.1738.v1.

[115] S. Ren, J. Li, K. Guo, and F. Li, “Medical video super-resolution based on asymmetric back-projection network with multilevel error feedback,” *IEEE Access*, vol. 9, pp. 17909–17920, 2021.

[116] K. Guo et al., “Video super-resolution based on inter-frame information utilization for intelligent transportation,” *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 11, pp. 13409–13421, Nov. 2023.

[117] M. V. Daithankar and S. D. Ruikar, “Adas Vision System With Video Super Resolution,” in *Autonomous Driving and Advanced Driver-Assistance Systems (ADAS): Applications, Development, Legal Issues, and Testing*. Boca Raton, FL, USA: CRC Press, 2021, pp. 135–148.

[118] X. Yang, W. Xiang, H. Zeng, and L. Zhang, “Real-world video super-resolution: A benchmark dataset and a decomposition based learning scheme,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 4761–4770.

[119] M. Jeelani, Sadbhawna, N. Cheema, K. Illgner-Fehns, P. Slusallek, and S. Jaiswal, “Expanding synthetic real-world degradations for blind video super resolution,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2023, pp. 1199–1208.

[120] A. G. Howard et al., “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” 2017, *arXiv:1704.04861*.

[121] M. Tan and Q. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.

[122] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” 2015, *arXiv:1503.02531*.

[123] E. Shechtman, Y. Caspi, and M. Irani, “Space-time super-resolution,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 4, pp. 531–545, Apr. 2005.

[124] Z. Geng, L. Liang, T. Ding, and I. Zharkov, “RSTT: Real-time spatial temporal transformer for space-time video super-resolution,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 17420–17430.

[125] X. Xiang, Y. Tian, Y. Zhang, Y. Fu, J. P. Allebach, and C. Xu, “Zooming slow-mo: Fast and accurate one-stage space-time video super-resolution,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3367–3376.

[126] G. Xu, J. Xu, Z. Li, L. Wang, X. Sun, and M.-M. Cheng, “Temporal modulation network for controllable space-time video super-resolution,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 6384–6393.

[127] M. Hu, K. Jiang, Z. Wang, X. Bai, and R. Hu, “CycMuNet+: Cycle-projected mutual learning for spatial-temporal video super-resolution,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 13376–13392, Nov. 2023.

[128] Z. Xiao, Z. Xiong, X. Fu, D. Liu, and Z.-J. Zha, “Space-time video super-resolution using temporal profiles,” in *Proc. 28th ACM Int. Conf. Multimedia*, 2020, pp. 664–672, doi: 10.1145/3394171.3413667.

[129] S. Dutta, N. A. Shah, and A. Mittal, “Efficient space-time video super resolution using low-resolution flow and mask upsampling,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 314–323.