IBC2023

TECHNICAL OVERVIEW OF RECENT AI/DL MODEL TRENDS FOR SUPER-RESOLUTION VIDEO ENHANCEMENT

Nelson Francisco

MediaKind, UK

Julien Le Tanou

MediaKind, France

ABSTRACT

High production costs have been a key factor in delaying a widespread deployment of UHD broadcast offerings. Only a few special events tend to be produced and broadcast on UHD, with most of the true 4K content coming from streaming providers such as Netflix, Amazon Video or Disney+, and even on those cases, content availability is still significantly more limited than for their non-4K assets. As a result, the potential of UHD displays is not fully exploited, with the final picture representation relying on the viewing device upscale capabilities, usually highly constrained by computational and power consumption limitations.

High-quality and reliable up-conversion can present a viable solution to accelerate UHD availability, allowing content providers to significantly reduce costs by complementing their offerings with high-quality content upscaled from existing HD libraries and leverage on their current production pipelines all the way up to the final up-conversion stage, while retaining control over how content is rendered on UHD screens. Widely investigated deep learning-based methods are perfect candidates for such applications, greatly outperforming traditional techniques and being particularly well suited for cloud deployments, where GPU acceleration can help providing high-throughput inference.

This paper provides a comprehensive overview of state-of-the-art deep learning-based super-resolution methods and their respective advantages and drawbacks, focusing on how they can be tailored for practical deployments in the cloud to mitigate their typical limitations.

INTRODUCTION

Super-resolution (SR) methods [1] refer to the process of generating high-resolution images or videos from low-resolution inputs. Such techniques have been an important topic of research for several decades, with early SR methods relying on spatial interpolation techniques [2,3]. While those methods were simple and effective, the quality of the upscaled images was constrained by their inability to generate high frequency details. Some progress was made over the years with the introduction of more complex approaches, including statistical, prediction-based, patch-based, or edge-based methods [4-16]. The most significant advances were however delivered by emerging deep learning techniques [17,18] and particularly convolutional neural networks (CNNs). Although Convolutional Neural Networks (CNNs) have been around since the 1980s, it wasn't until the mid-1990s that they started to gain widespread attention in the research community

[20], mainly due to the lack of hardware suited to train and run sizeable networks. CNNs have since undergone numerous improvements and became one of the most powerful and widely used deep learning techniques for image analysis and processing tasks. In recent years, CNNs have achieved state-of-the-art performance in tasks ranging from image classification [21,22], object detection [23], or semantic segmentation [24], among many others [25].

The first convolutional neural network (CNN) based super-resolution method is generally attributed to Dong et al., who proposed the "SRCNN" (Super-Resolution Convolutional Neural Network) in their 2015 paper “Image super-resolution using deep convolutional networks” [26]. The authors developed a three-layer CNN architecture able to learn the mapping from low-resolution to high-resolution images by using a large training dataset. Numerous CNN-based super-resolution methods followed, each improving in areas such as the data mapping, networks architecture and size, optimization function or computational efficiency, with many of those methods achieving state-of-the-art performance on various benchmark datasets over the years [27,31].

Another crucial development was delivered with the inception of Residual Networks [32]. In a traditional deep neural network, as the number of layers increases, gradients become weaker and weaker during the training process as they are propagated back through the network. Some of these gradients may vanish or explode, causing instability or stopping the learning process from converging. This made it increasingly challenging to train very deep networks. The ResNet architecture tackles the issue by introducing the concept of residual connections, where the output of some layer can bypass others to be directly added to the input of a subsequent layer. This allows the network to learn residual mappings rather than full mappings, making it possible to train significantly deeper networks that can often reach hundreds of layers. This made the ResNet architecture highly popular for many computer vision tasks, including super-resolution.

Building up in those innovations and in the increase of hardware capabilities to train and run larger and more complex networks, the super-resolution field has been evolving very quickly over the past years. Advances in generative models such as Auto-Encoders and Generative Adversarial Networks (GANs) opened new possibilities, providing high-quality upscales that match the underlying distribution of high-resolution images even in cases where the input data is noisy or incomplete. New trends such as transformer models and diffusion are still pushing the boundaries of what can be achieved even further.

However, each network architecture comes with its own advantages and drawbacks, so it becomes of great importance to tailor each solution to its target application, especially since the balance between computational complexity and performance is often the most important constraint in a practical system’s design.

DEEP LEARNING SUPER-RESOLUTION METHODS

While both the input and output of a Single Image Super-Resolution (SISR) algorithm are individual images, Video Super-Resolution (VSR) algorithms must generate multiple high-resolution frames from multiple low-resolution frame inputs. A trivial approach for VSR is to apply a SISR algorithm to each of the input frames, but such approach usually introduces artifacts such as flickering or shimmering due to inconsistencies in the details generated for each output frame. VSR methods need to maintain temporal consistency to maximize perceptual quality, and this is typically achieved by using multiple frames of the low-resolution input to generate each upscaled frame for the upscaled video. Feature

alignment on input frames is commonly achieved by using motion compensation, optical flow, or other similar methods [33], resulting in algorithms computationally more complex than equivalent SISR algorithms.

Despite this fundamental difference, SISR and VSR share similar network architectures, with algorithms still falling into the same classes. For this reason and for the sake of simplicity, we will focus on the analysis of SISR algorithm, but comparative results and relative merits and drawbacks of each class of algorithms can be extrapolated for VSR solutions.

PSNR Oriented methods

PSNR-oriented methods are trained with simple distribution assumption-based losses [26], being able to achieve excellent PSNR [34-37], but often resulting in smooth images with a lack of detail. During the training process, patches of high-resolution images are downscaled and used as input of a super-resolution generator network, which upscale them back to the original resolution. The original images are then used as the ground-truth, so that a loss between the original and the upscaled patches can be calculated. The network coefficients are trained backpropagating the loss function gradients to minimize the error between the super-resolution upscales and the originals.

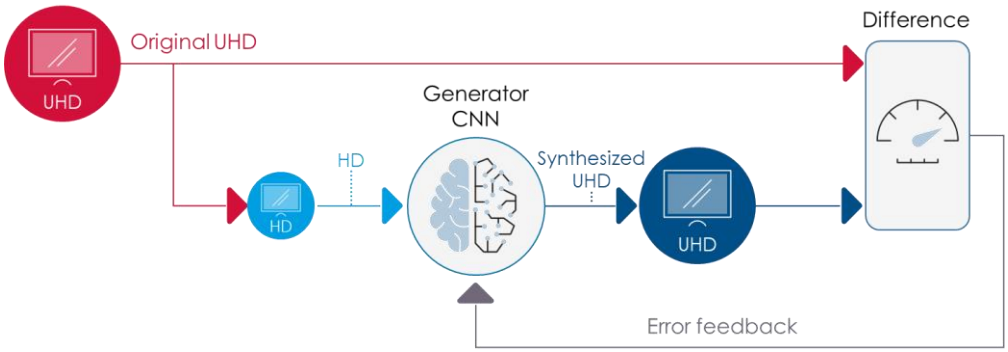

Figure 1 - PSNR-based method approach

The ill-nature of the problem means however that there are multiple solutions possible while mapping a low resolution into a high-resolution patch, and minimizing reconstruction losses tends to favour a prediction resulting from averaging all plausible HR solutions, leading to the significant reduction of high-frequency details

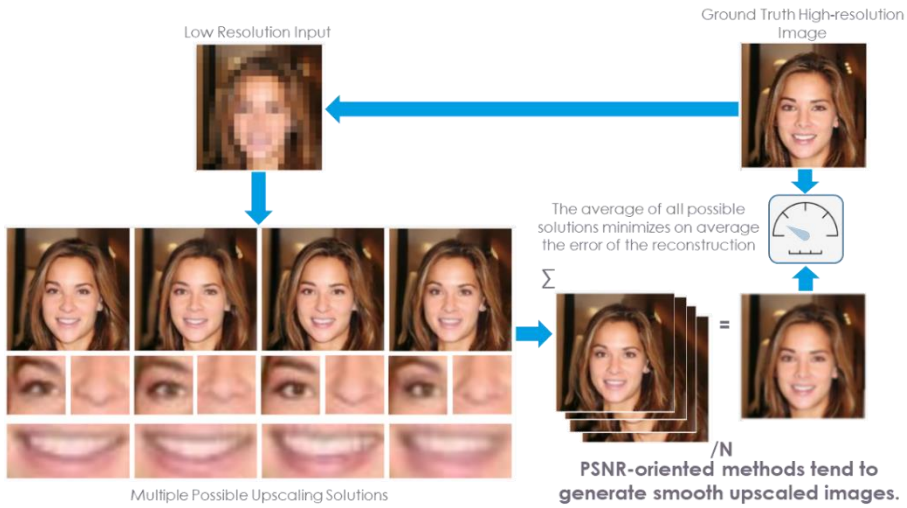

Figure 2 - Ill-posed nature of the PSNR oriented optimization

A synthetic example is presented in Figure 2, where the low-resolution input could be mapped in any of the 4 possible HR faces, which are all very similar and equally credible, and potentially presenting identical MSE, but which present slightly different features (as can be seen in the detail crops). Assuming the network is presented with the LR image and all of the 4 HR images are equally correct answers for the problem, the network will be tempted to produce an upscaled output resulting from the average of all the possible faces, as this would, in average, minimize the loss to the ground truth if all the 4 faces are randomly presented to the network. This results in a feature blending and a consequent smoothed-out reconstruction, which is not necessarily what we aim to achieve to maximize the perceptual quality of the up-sampled image. In a real application scenario, the original high-resolution image won't even exist for comparison, so the generated detail just needs to look credible and consistent for the super-resolution image to be considered of high quality by viewers.

Generative Adversarial Networks

Generative Adversarial Networks (GANs) were originally proposed by Goodfellow in 2014 [39] and later successfully used for super-resolution applications [39,40]. GANs try to address the over-smoothing problem of PSNR-based methods by replacing the simple loss function by a complementary CNN, trained to rank the credibility of the upscaled images. For that purpose, the discriminator is alternatingly presented with original ground-truth and upscaled patches coming from the generator, learning to determine the likelihood of a given high-resolution patch being original or synthetic.

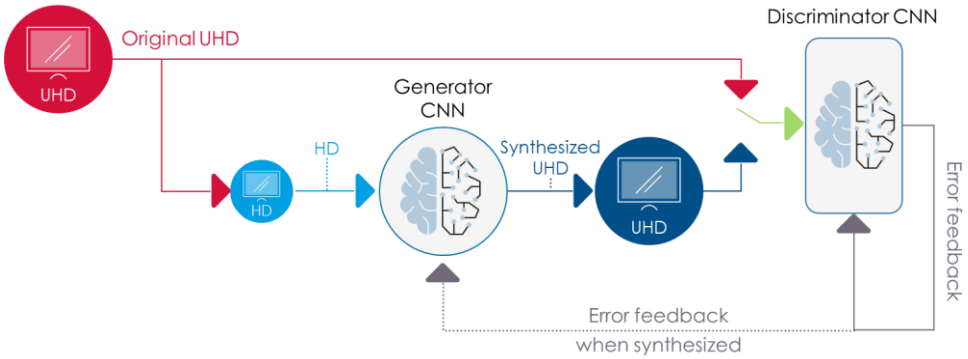

Figure 3 - GAN based super-resolution

By training both networks simultaneously, they thrive on each other's success: while the discriminator gets better at distinguishing between original and synthetic high-resolution patches, the generator must become better at generating more credible high-resolution upscales to successfully trick the generator. Analogously, the better the generator becomes at creating synthetic high-resolution images, the more accurate the discriminator must become at distinguishing them from the ground truth high-resolution images.

GANs provide a significant step up in the perceptual quality of the upscaled images, generating sharper pictures with richer high frequency details, but pose some additional challenges related to their design and training. First, they are intrinsically more computationally expensive to train and result in a larger memory footprint, given 2 NN must be trained concurrently. However, since the discriminator is not used during inference, this issue is mitigated in practical applications. Second, instead of optimizing the generator to minimize a well-defined metric (loss-function), the optimization function in a GAN is itself varying as the discriminator progressively learns. This make GANs prone to

suffer from mode collapse, essentially a situation where the generator "collapses" to producing only a subset of the target distribution, rather than the full distribution.

Several factors contribute to mode collapse, such as the discriminator being too powerful when compared to the generator or simply not accurate enough for the application. While in the first case, the generator gets trapped on local minima as it struggles to produce diverse samples that can fool the discriminator (this can also happen if the training data is too limited so that the generator fails to learn the data full distribution), on the second case, the generator fails to produce varied, high-quality outputs since the fact it can trick the discriminator to believe a patch is genuine every time means it has no incentive further improve or diversify its outputs. Mode collapse is also likely to occur if the learning rates for the 2 networks are not properly balanced and one of the networks converges much quicker than the other.

Researchers have developed various techniques that complement careful training data selections [41] and learning rate tuning in reducing the likelihood of mode collapse. Those include modifying the loss function [40], tuning the architectures of both the generator and discriminator [42], and adding regularization terms to the model [43]. Figure shows an example were a weighted combination of a simple loss with a discriminator network helps providing the benefits of a GAN approach while mitigating some of the risks.

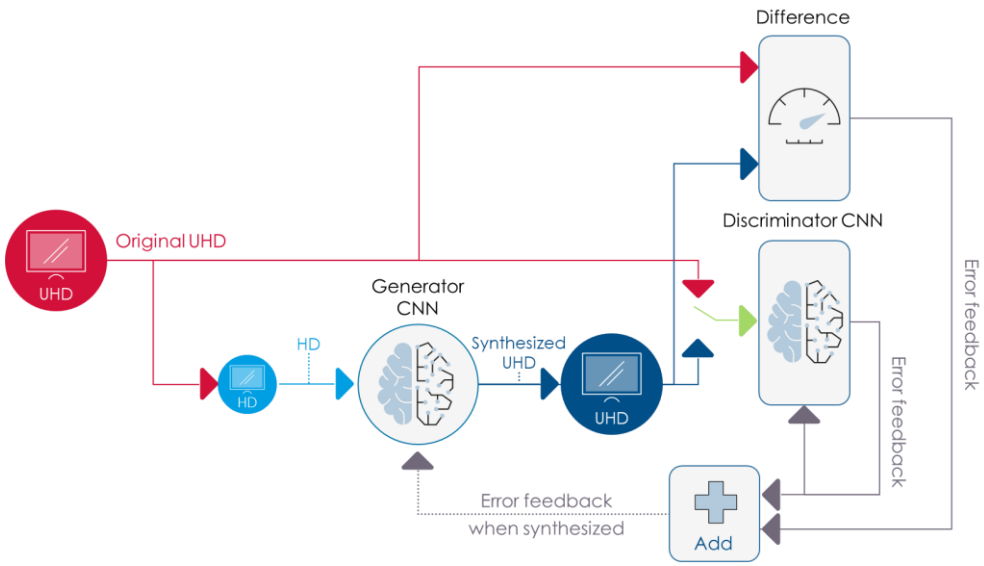

Figure 4 - Gan with hybrid loss function

The hybrid approach also helps controlling mode hallucination and avoid generating high resolution upscales that although convincing may not completely correlate to the input. It is not uncommon to observe histogram shifts in the upscaled images that impact objective metrics of upscales generated by GANs, being possible to reduce its likelihood with the addition of a simple loss term. A careful design and tuning of the network architecture and the addition of residual layers may also help mitigating the issue.

Transformer Models

Transformer models [44] are a type of neural network architecture originally developed for natural language processing (NLP) tasks but has since also been successfully applied to other types of sequence-based data, such as images [45], video [46] and audio signals [47]. The main building blocks of transformer models are self-attention mechanisms and multi-layer feedforward networks.

In a typical transformer, the input data is first embedded into a sequence of vectors, which are then processed by a stack of identical layers. Each layer consists of two sub-layer types: a self-attention mechanism and a multi-layer feedforward network. The self-attention mechanism allows the model to learn a global representation of the input sequence by attending to different parts of the sequence at different levels of granularity. At each position in the sequence, the self-attention mechanism calculates a weighted sum of the other positions in the sequence, where the weights are determined by a learned attention function. This weighted sum is then used to compute a new representation of the current position, which is passed to the next layer. The multi-layer feedforward network applies a non-linear transformation to the self-attention output at each position in the sequence, which helps the model to capture more complex relationships between different parts of the sequence. The output of the feedforward network goes then through a residual connection and a layer normalization operation before being passed to the next layer. After the input sequence has been processed by the stack of transformer layers, the final output goes through linear layers to generate the model's prediction for the task at hand.

A few adjustments are applied to the transformer model to process image data. Typically, images are first processed by a convolutional neural network (CNN) to extract a set of feature maps that represent the low-level visual features. Feature maps are then split into non-overlapping patches, which are flattened and mapped to continuous vectors, which are then combined with learned positional embeddings to be processed by the multi-head self-attention mechanism.

The learned embeddings and self-attention mechanism effectively map the image data into a latent space, and since the optimization is performed in that domain, transformers can mitigate some of the issues related to the ill-nature of the problem, meaning outputs are less likely to suffer from smoothness than for PSNR-methods. Since optimization relies in a single loss, transformer models do not suffer from mode collapse, being easier to converge than GANs. They tend however to be relatively large and require vast amounts of computation and memory to run.

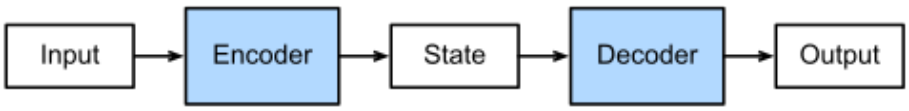

Figure 5 - Transformer domain mapping

In [48], the authors proposed to adapt the popular Swin Transformer architecture [49] to be used in a generic video restoration algorithm named SwinIR. Swin is a variant of the transformer that uses hierarchical feature representation and shift windows to capture spatial information. Although SwinIR performed well on several image restoration tasks, it tends to generate smooth images that lack realism when using for super-resolution.

In [50], the authors proposed to use a transformer with hierarchical patches and in [51], the authors proposed to use the SwinV2 [52] transformer, an updated version of the original Swin transformer that incorporates several innovations and improvements. Those include cross-window aggregation, allowing larger context modelling, residual connections in the normalization layer that improve stability during training, a multi-scale attention that aggregates features at multiple scales to capture both local and global information, layer-wise scaling to balance the contribution of each layer to the final output and prevent gradient explosion or vanishing and the use of depth wise convolution in the stem layer to reduce the number of parameters and improve the speed of the inference process. Overall, this allowed Swin2R to generate more credible super-resolution images when

compared with the original SwinIR, but overall, the algorithm still tends to suffer from some over-smoothing when compared to other techniques.

Overall, successive improvements in the transformer architecture allowed those methods to achieve competitive results for super-resolution application, but at the cost of a relatively high training and inference costs.

Flow Models

While other approaches try to learn a deterministic mapping between the low and high-resolution pairs, flow-based methods [53] directly account for the fact that any given low-resolution image can effectively be mapped into infinite compatible high-resolution images, by aiming to instead capture the full distribution of natural high-resolution images conditioned to their corresponding low-resolution counterparts [54].

An invertible NN that maps HR-LR image pair to a latent variable is used to parametrize the conditional distribution function, with the bijection between latents and data meaning any given high-resolution image can always be exactly reconstructed from the latent space. This allows to train the NN over a large dataset of high-resolution and low-resolution pairs using a single negative log-likelihood loss.

The use of a single loss allows Flow-based methods to avoid mode collapse and other training instabilities but results in extremely large footprints and high training costs due to the strong architectural constraints to keep the bijection between latents and data.

Overall, flow model present some of the most balanced results between image sharpness, objective error metrics and level of perceived artifacts, but at the expense of complexity and memory requirements that may make them unsuitable for many applications.

Diffusion Models

Diffusion probabilistic models [54] are another type of generative models well suited to tackle problems with one-to-many solutions like super-resolution. They have then been successfully used for super-resolution [55,56], as well as other applications such as speech [57] and image synthesis [58].

Diffusion probabilistic models rely on the use of a Markov chain to convert data x_0 into a latent variable x_T with a simple distribution (such as a Gaussian distribution), by gradually adding noise ε in a diffusion process, and then predict the noise ε in each diffusion step to recover the data x_0 through a learned reverse process.

There are then 2 stages in a diffusion model: a Forward Diffusion stage, where the image is corrupted by gradually introducing noise until it becomes complete random noise (left to right in Figure 6), and the reverse process, where a series of Markov Chains are used to recover the data from the Gaussian noise by gradually removing the predicted noise at each time step (right to left in Figure 6).

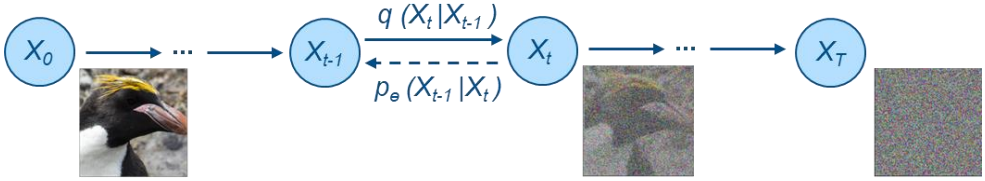

Figure 6 - Diffusion process

Super-resolution diffusion models use a low-resolution image as an input to condition the noise distribution and propagate information through a diffusion process to generate a high-resolution image or the high frequency detail to add to an upscaled image. In this process, the low-resolution image is transformed into a higher-dimensional space where it is easier to recover fine-grained details, using a learned function that maps the low-resolution input to the higher-dimensional space.

Once the input is in the higher-dimensional space, a diffusion process is applied to propagate information across the dimensions. This process can be thought of as a random walk, where each pixel in the low-resolution image corresponds to a particle that moves through the high-dimensional space, influenced by neighboring particles. As the particles move, they exchange information, allowing fine-grained details to be recovered.

After the diffusion process is complete, the resulting high-resolution image is generated by mapping the particles back to the original image space. This is achieved using a learned function that maps the high-dimensional representation to the high-resolution output.

Diffusion super-resolution models are typically trained using pairs of low- and high-resolution images, with the goal of learning to accurately recover fine-grained details from the low-resolution input, through the use of a loss function that penalizes differences between the generated SR image and the ground truth HR image. By training the diffusion models by optimizing a variant of the variational lower bound, diffusion probabilistic models also avoid the mode collapse problems encountered by GANs.

Diffusion models can achieve state-of-the-art results for super-resolution application by relying in smaller models that typically are cheaper to train when compared to flow or transformer models. However, their iterative nature means multiple inference passes are required to generate each output image, resulting in relatively high inference costs. This makes diffusion models especially suited for applications where memory resources are limited but inference time is not critical, but may prevent its applicability for applications where there are strong constraints in inference time. The iterative nature of Diffusion models also has the benefit of scalability, as it is possible to run less iterations and accept less quality when limited resources are available and increase the number of iterations to improve quality when resources become available.

METHOD COMPARISON

In order to compare the different super-resolutions architectures discussed in the previous section in terms of quality and computational complexity, the same set of images was upscaled using a representative method from each class of algorithms. We used the DIV2K dataset [27], a publicly available dataset with 900 high-resolution RGB images comprising a large diversity of contents. The DIV2K dataset is divided into a training subset with 800 images and a validation subset with 100 images, and since the training subset has been used by most authors to train their models, only the 100 validation images were used for the purpose of this evaluation.

Although exceeding what is required for what can be assumed to be the most typical applications on video delivery (such as SD to HD or HD to 4K), we used an upscale factor of 4x in both the horizontal and vertical directions to emphasise the difference between the various methods and highlight their most typically introduced artifacts.

The input LR images we generated by down-sampling each of the original HR images in the DIV2K validation set using a bicubic kernel [2]. Those LR images were then upscaled to SR images using each one of the methods in analysis, so that quality metrics between

the original ground truth (HR) and the upscaled (SR) images could be calculated. Besides the widely used PSNR and SSIM [59], we also computed the LPIPS [60], a DL reference-based image quality evaluation metric that computes the perceptual similarity between the ground truth and the SR images and is highly effective at assessing the amount of high frequency detail introduced by the upscaling process. To further evaluate the consistency and correlation between the SR and LR images, we also include the LR-PSNR [54] values, computed as the PSNR between the down-sampled SR image and the input LR image.

A Bicubic upscaling interpolation [2] was used as the reference for this comparison, and RRDB [40] was adopted as the representative PSNR-oriented method. RRDB uses the same generator network used in ESRGAN, the method adopted to illustrate the performance of GAN methods. The only difference between both methods is that while the former is trained by using a simple L1 loss, the latter combines the simple loss with the result of a discriminator network to calculate an adversarial loss. This allows to directly evaluate the benefits of a GAN architecture by comparing head-to-head two identical NN-trained with and without an adversarial network. Swin2R [51] was used to represent transformer methods, SRFlow [54] to illustrate the performance of flow-based models and SRDIFF [56] to represent diffusion super-resolution models, since to our knowledge, there is no complete open-source implementation of Google's SR3 [55]. For all the methods, the default architecture proposed by the authors in their original papers was used, as well as the pre-trained models they provide. A Nvidia RTX3080Ti GPU with 12Gb of VRAM was used for inference, with the results are summarized in Table 1.

Method	PSNR ↑	SSIM ↑	LR-PSNR ↑	LPIPS ↓	Network parameters ↓	Memory Usage [Mb] ↓	Throughput ↑
Bicubic	26.693	0.766	38.697	0.421	-	-	-
RRDB	29.475	0.844	53.737	0.262	16M	6213	2.1 fps
ESRGAN	26.636	0.764	42.613	**0.119**	16M	6213	2.1 fps
SRFLOW	27.109	0.756	52.066	0.124	40M	9895	0.8 fps
SWIN2R	**29.621**	**0.848**	**54.605**	0.256	12M	7165	1 fps
SRDIFF	27.125	0.785	49.617	0.132	13M	9101	3 it/s. 100 iterations per image 31s image
Our model	26.775	0.7687	50.844	0.127	**1.6M**	**5689**	**6 fps**

Table 1 - Experimental results

When comparing RRDB with the bicubic interpolation, it can be observed that every metric is consistently improved (higher is better for all metrics except LPIPS where lower is better), clearly demonstrating the superiority of DL-based methods over interpolation methods. SWIN2R provides another step up over RRDB by again improving all the measured quality metrics, at the expense of higher computational complexity.

However, when analysing the LPIPS score in particular, it can be observed that although the RRDB and SWIN2R perform much better than the bicubic interpolation, they still fall considerably behind the remaining methods. This correlates directly with what can be observed on the example shown in Figure 7, with the images upscaled using those 2

methods being noticeably softer than the ones upscaled using ESRGAN, SRFLOW and SRDIFF.

ESRGAN achieves the best LPIPS of all methods in the comparison, suggesting its upscaled images are richer on high frequency details. This aligns once again with what can be observed in Figure 7, demonstrating the effectiveness of LPIPS to assess the quality of super-resolution algorithms and in particular the level of detail introduced in the upscaled images. The leading LPIPS achieved by ESRGAN comes however at the expense of some degradation in the other metrics when compared to other methods, particularly when looking at the LR-PSNR. Hallucinated details often do not consistently match those in the ground-truth image, but this may not be a significant problem in real world application if the detail is visually credible, considering the viewers won't have access to the ground truth.

Both SRFLOW and SRDIFF achieve comparable results, with a good balance between metrics and perceived sharpness.

Figure 7 - Detail from image #830 from the DIV2K dataset

In the last row of Table 1, we present the results for our proprietary method, that has been developed inhouse. It uses a GAN architecture but with several tweaks and improvements over other proposals in the literature, including ESRGAN. Improvements allowed to significantly reduce computational complexity while maintaining competitive results, and most notably, changes in the residual network structure improved the correlation of upscaled images with the ground truth. This is reflected in a much-improved LR-PSNR scores when compared to ESRGAN, being now competitive with all other methods. Improvements that contributed to the increase in efficiency include multiple adjustments in residual structure and other aspects of the network architecture, a tuned hybrid loss that mitigates the potential for network hallucination, a more efficient mapping between LR and HR layers which allowed reducing the number of network layers without heavily affecting the quality of the output, and a carefully tuned training strategy that reduced the probability of mode collapse. While combined, these changes allowed a 90% reduction in the number of network weights, with a direct impact in the computational complexity of the upscaling process, while improving all metrics relatively to ESRGAN, except for a small degradation

in the LPIPS score. Note that for the purpose of this comparison, our method, originally developed for video super-resolution and that supports simultaneous deinterlacing and upscaling was adapted for SISR and trained in the DIV2K dataset.

In the last column of Table 1, we include the throughput achieved with the RTX3080Ti GPU with each method. This allows estimation of the relative upscaling costs for each method, when running in a single GPU instance in the cloud. As expected, a strong correlation between the number of NN coefficients and the throughput can be observed, as the number of coefficients will obviously impact the number of multiply and add operations to be performed. The relationship is however not completely linear as other factors related to the NN architecture and bandwidth constraints affect the practical inference times.

As a reference for operational costs, a cloud instance with a single V100 GPU has a default cost of around $3/hour on the major cloud vendors, with the inference times approximately 30% slower than achieved with the RTX3080Ti. Performing a frame-by-frame upscale of 1 hour of 1920x1080 progressive 60fps video to 3840x2160 takes around 16h using our method, 47h using the default ESRGAN, 100h using SWIN2R, 125h using SRFLOW and a staggering 3300h to run the 100 iterations for each frame using SRDIFF, as proposed by the authors. This clearly demonstrates how operational costs can escalate quickly, with our method coming at around $50/h, the original ESRGAN at around $140/h, SWIN2R $300/h, SRFLOW $375/h and SRDIFF $10000/h, a cost perhaps only justifiable to high value assets such as movies. Note that those are only indicative costs, since the algorithms may have some scope for optimization over the models provided by the authors. Furthermore, we are only focusing on processing costs, not accounting for storage and ingress and egress required for a real-world application.

The results demonstrate that with careful design and optimization to mitigate some of the common issues often associated with GANs (such as mode collapse and hallucination), GAN-base methods can deliver competitive results with lower computational cost than some newly emerging approaches, despite their decline in popularity. They are also well suited for video upscale application as the cost of training the additional discriminator becomes irrelevant as the model is used to upscale large numbers of images.

Table 2 provides a relative comparison between the various types of architectures.

Type	Training complexity/cost	Memory Footprint	Inference complexity/ cost	Quality
PSNR-oriented	Low	Low	Low	Low. Images are smooth.
GAN	Medium	Low	Low	Medium. Images tend to be very sharp but may suffer from hallucination artifacts.
Transformers	High	High	High	Medium. Images can be smooth and look unnatural, somewhat cartoonish.
Flow based	High	High	High	High. Better overall balance between sharpness, consistency and metrics.
Diffusion	Low	Low	Very high (multiple iterations)	High. Better overall balance between sharpness, consistency and metrics.

Table 2 - Method comparison summary

SYSTEM ARCHITECTURE AND APPLICATIONS

Nvidia [61] and Microsoft [62] recently released their AI-driven super-resolution solutions that leverage on the local resources found on modern consumer-grade GPUs to perform real-time video upscaling and enhancement prior to display (Figure 8), showing how this technology is gathering interest among an increasing number of companies. Although these solutions are particularly well suited for applications where the stream is only accessed by a small number of viewers, such as video-conferencing and old legacy content streaming, or for when the bandwidth available for transmission is highly constrained, they present some drawbacks that can limit their success for other applications.

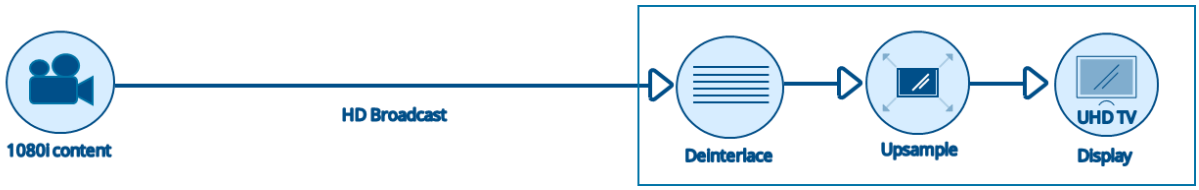

Figure 8 - Upscaling at the viewer ewer side

First, the video quality they can deliver is inherently limited as they rely in the viewing device hardware resources. Although modern consumer grade GPUs offer high computing capabilities for NN inference and generous amounts of memory, the fact upscaling must happen during real-time playback highly restricts the size and topology of the DL upscaling algorithm used. The results will also be dependent on the resources available on each device, meaning content providers have no control over how the video is effectively rendered. This directly limits the potential for content monetization as a premium video quality doesn't depend on the content provider but on the viewing device capabilities. Additionally, it is extremely energy inefficient at scale when the stream is decoded by multiple viewers, as the power-hungry upscaling will have to run on every compatible viewing device displaying the content.

For applications where the same stream is to be accessed by a large number of viewers, there are multiple advantages in performing the upscaling at the content provider side (Figure 9).

Figure 9 - Upscaling at the content provider side

First, the content provider retains control over how the content is rendered, guaranteeing high quality to every supporting device irrespective of it processing capabilities. This may allow the content provider to monetize the higher quality content, while the upscale can rely on more resources as it no longer depends on the viewing devices capabilities. Heavy resource usages are quickly offset by the fact that the high quality upscale only needs to be performed once prior to distribution. Cloud solutions are particularly well suited for this application, by providing vast amounts of hardware accelerated resources with fully scalable costs. The commercial value of the content becomes the main driver in determining the amount of resources economically viable for the application on hand,

helping to determine the upscaled video perceptual quality operating point. Figure 10 schematizes how multiple VM can be used to increase the throughput of the upscaling process.

The main drawback of performing the upscaling at the content provider side is the fact it will likely increases the bandwidth required for the content distribution, a factor that can however be mitigated with an efficient CDN design, especially when the content is being accessed by a large number of viewers.

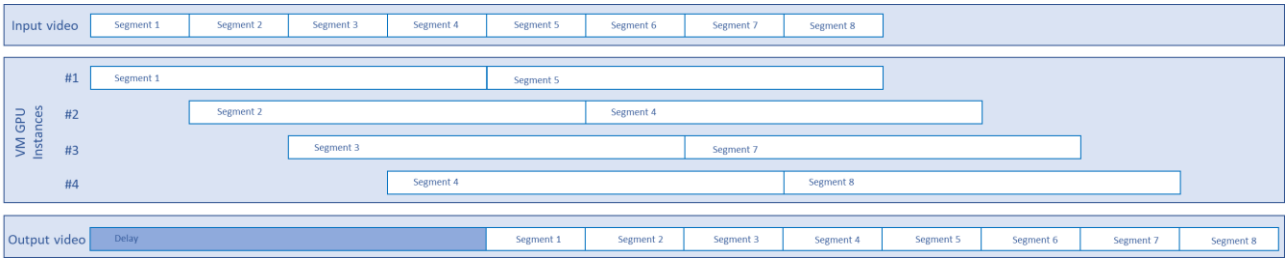

Figure 10 - VM instance parallelization

Ultimately, the nature of the application and content value will be the most important factors in determining the best DL upscaling approach. Decoder side upscaling will likely be the best solution for streams with a very low number of viewers, such as video conference or old legacy content, not only because of the limited monetization potential but also because it guarantees the upscaling cost is only incurred when needed. Small footprint, low inference complexity methods which do not necessarily deliver state-of-the-art results are likely to be a good option on those cases. Very high value offline content such as movies, will likely benefit from the best possible upscaling irrespectively of cost, as the high value of the asset will justify the upscaling costs and the processing time is unlikely to be critical. Such methods can rely in some of the most complex, highly tailored, and better performing approaches, based on state-of-the art methods such as flow and diffusion models, running on the cloud, where processing can be scaled accordingly. High value live content such as sports will likely impose some restrictions on processing power since the upscaling has to happen in real-time and additional delay is critical, but cloud scaling may help providing the required processing power withing the budgetary constraints, as illustrated in Figure 10. Finally, for any other type of content, a well performing and scalable solution may help achieving the best possible video quality on the processing resources available, aiming for the right balance of what is economically viable given the content value.

The success of the upscaling strategy is also highly dependent on the dataset used to train the model [63], so making sure it is adequately varied and representative of what is going to be processed in real-life is paramount. Most of the models in the literature have been trained and evaluated using pristine images and video, but in real-world applications, input images will often contain artifacts from previous compression stages that need to be modelled and taken into consideration [64,65], under risk of producing poor quality upscales. Ultimately, it is pointless to adopt a very complex upscaling algorithm that is not resilient enough to the variations in the input quality, as it may perform worse than a simpler but more reliable approach.

CONCLUSIONS

In this paper, we provided an overview of some of the most promising trends and architectures for video super-resolution enhancement, comparing their relative performance in terms of objective metrics, perceptual quality, and computational

complexity. We focused our analysis on SISR as this allowed us to dissociate the effect of the spatial detail enhancement achieved by each NN architecture, from the underlying success of the feature alignment strategy used by each VSR solution [66-71].

Super-resolution methods have come a long way since their inception in the 1980s, with advances in computer vision, machine learning, and deep learning making it possible to generate high-quality images and videos from low-resolution inputs. While the field of SR methods is still evolving, recent developments in generative models are likely to play a significant role in shaping of content distribution, helping content providers to lower production costs to increase their UHD offerings.

REFERENCES

[1] S. C. Park, M. K. Park, and M. G. Kang, “Super-resolution image reconstruction: a technical overview,” IEEE Signal Processing Magazine, vol. 20, no. 3, pp. 21–36, 2003.

[2] R. Keys, “Cubic convolution interpolation for digital image processing,” IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 29, no. 6, pp. 1153–1160, 1981.

[3] C. E. Duchon, “Lanczos filtering in one and two dimensions,” Journal of Applied Meteorology, vol. 18, no. 8, pp. 1016–1022, 1979.

[4] W. T. Freeman, T. R. Jones, and E. C. Pasztor, “Example-based superresolution,” IEEE Computer Graphics and Applications, vol. 22, no. 2, pp. 56–65, 2002.

[5] H. Chang, D.-Y. Yeung, and Y. Xiong, “Super-resolution through neighbor embedding,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 275–282, 2004.

[6] M. Aharon, M. Elad, A. Bruckstein et al., “K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation,” IEEE Transactions on Signal Processing, vol. 54, no. 11, p. 4311, 2006.

[7] A. Marquina and S. J. Osher, “Image super-resolution by TV regularization and Bregman iteration,” Journal of Scientific Computing, vol. 37, no. 3, pp. 367–382, 2008.

[8] J. Sun, Z. Xu, and H.-Y. Shum, “Image super-resolution using gradient profile prior,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[9] S. Dai, M. Han, W. Xu, Y. Wu, Y. Gong, and A. K. Katsaggelos, “Softcuts: a soft edge smoothness prior for color image super-resolution,” IEEE Transactions on Image Processing, vol. 18, no. 5, pp. 969–981, 2009.

[10] J. Yang, J. Wright, T. S. Huang, and Y. Ma, “Image super-resolution via sparse representation,” IEEE Transactions on Image Processing, vol. 19, no. 11, pp. 2861–2873, 2010.

[11] Q. Yan, Y. Xu, X. Yang, and T. Q. Nguyen, “Single image superresolution based on gradient profile sharpness,” IEEE Transactions on Image Processing, vol. 24, no. 10, pp. 3187–3202, 2015.

[12] R. Timofte, V. De, and L. Van Gool, “Anchored neighborhood regression for fast example-based super-resolution,” in Proceedings of the IEEE international Conference on Computer Vision, 2013, pp. 1920–1927.

[13] S. Schulter, C. Leistner, and H. Bischof, “Fast and accurate image upscaling with super-resolution forests,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp.3791–3799.

[14] F. Cao, M. Cai, Y. Tan, and J. Zhao, “Image super-resolution via adaptive p (0 < p < 1) regularization and sparse representation,” IEEE Transactions on Neural Networks and Learning Systems, vol. 27, no. 7, pp. 1550–1561, 2016.

[15] W. Yang, Y. Tian, F. Zhou, Q. Liao, H. Chen, and C. Zheng, “Consistent coding scheme for single-image super-resolution via independent dictionaries,” IEEE Transactions on Multimedia, vol. 18, no. 3, pp. 313–325, 2016.

[16] J. Liu, W. Yang, X. Zhang, and Z. Guo, “Retrieval compensated group structured sparsity for image super-resolution,” IEEE Transactions on Multimedia, vol. 19, no. 2, pp. 302–316, 2017.

[17] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 8, pp. 1798–1828, 2013.

[18] J. Schmidhuber, “Deep learning in neural networks: An overview,” Neural Networks, vol. 61, pp. 85–117, 2015.

[19] Y. LeCun, Y. Bengio and G. Hinton. “Deep learning”. Nature, 521(7553), pp.436-444, 2015.

[20] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.

[21] A. Krizhevsky, I. Sutskever, G. Hinton, “ImageNet classification with deep convolutional neural networks”. Advances in neural information processing systems, 25, 1097-1105, 2012.

[22] K. He, X. Zhang, S. Ren, J. Sun, “Deep residual learning for image recognition”. Proceedings of the IEEE conference on computer vision and pattern recognition, 770-778, 2016.

[23] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, “You only look once: Unified, real-time object detection”. Proceedings of the IEEE conference on computer vision and pattern recognition, 779-788, 2016.

[24] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs”. IEEE transactions on pattern analysis and machine intelligence, 40(4), 834-848, 2018.

[25] J. Wang, Z. Li, “Research on Face Recognition Based on CNN”. IOP Conference Series: Earth and Environmental Science, 2018. 170. 032110. 10.1088/1755-1315/170/3/032110.

[26] C. Dong, C. Loy, K. He, X. Tang. “Image super-resolution using deep convolutional networks”. IEEE transactions on pattern analysis and machine intelligence, 38(2):295–307, 2015.

[27] R. Timofte, S. Gu, J. Wu et al., “NTIRE 2018 challenge on single image super-resolution: methods and results”. In CVPR Workshops, 2018.

[28] J. Cai, S. Gu, R. Timofte et al., “NTIRE 2019 challenge on single image super-resolution: methods and results”. In CVPR Workshops, 2019.

[29] A. Lugmayr, M. Danelljan, R. Timofte, et al., “NTIRE 2020 challenge on single image super-resolution: methods and results”. In CVPR Workshops, 2020.

[30] S. Son, S. Lee, S. Nah, R. Timofte et al., “NTIRE 2021 challenge on single image super-resolution: methods and results”. In CVPR Workshops, 2021.

[31] R. Yang, R, Timofte, M. Zheng Q. Xing, “NTIRE 2022 Challenge on Super-Resolution and Quality Enhancement of Compressed Video: Dataset, Methods and Results”. In CVPR Workshops, 2021.

[32] K. He, X. Zhang, S. Ren, J. Sun, “Deep Residual Learning for Image Recognition”. arXiv preprint arXiv:1512.03385, 2016.

[33] S. Shi, J. Gu, L. Xie, X. Wang, Y. Yang, C. Dong, “Rethinking Alignment in Video Super-Resolution Transformers”. arXiv preprint arXiv:2207.08494, 2022.

[34] B. Lim, S. Son, H. Kim, S. Nah, K. Lee. “Enhanced deep residual networks for single image super-resolution”. In CVPRW, pages 136–144, 2017.

[35] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, Y. Fu, “Image super-resolution using very deep residual channel attention networks”. In ECCV, pages 286–301, 2018.

[36] Y. Qiu, R. Wang, D. Tao, J. Cheng. “Embedded block residual network: A recursive restoration model for single-image super-resolution”. In ICCV, pages 4180–4189, 2019.

[37] Y. Guo, J. Chen, J. Wang, Q. Chen, J. Cao, Z. Deng, Y. Xu, M. Tan. “Closed-loop matters: Dual regression networks for single image super-resolution”. In CVPRW, pages 5407–5416, 2020.

[38] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in Proceedings of the Advances in Neural Information Processing Systems, 2014, pp. 2672–2680.

[39] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al., “SRGAN: Photo-realistic single image super-resolution using a generative adversarial network”. CVPR, 2017.

[40] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, C. Loy, Y. Qiao, X. Tang, “ESRGAN: Enhanced super-resolution generative adversarial networks”. ECCV, 2018.

[41] X. Wang, L. Xie, C. Dong, Y. Shan. “Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data”. In ICCVW, 2021.

[42] W. Zhang, Y. Liu, C. Dong, Y. Qiao, “RankSRGAN: Generative adversarial networks with ranker for image super-resolution”. In ICCV, 2019.

[43] M. Cheon, J. Kim, J. Choi, J. Lee, “Generative adversarial network based image super-resolution using perceptual content losses”. Proceedings of ECCV, pages 0–0, 2018.

[44] A. Vaswani, N. Shazeer, N. Parmar et al., “Attention Is All You Need”. arXiv preprint arXiv:1706.03762, 2017.

[45] Z. Lu, J. Li, H. Liu, C. Huang, L. Zhang, T. Zeng, “Transformer for Single Image Super-Resolution”. arXiv preprint arXiv:2108.11084, 2022.

[46] J. Cao, Y. Li, K. Zhang, J. Liang, L. Van Gool, “Video Super-Resolution Transformer”, arXiv preprint arXiv:2106.06847, 2021.

[47] P. Verma, J. Berger, “Audio Transformers:Transformer Architectures For Large Scale Audio Understanding. Adieu Convolutions”. arXiv preprint arXiv:2105.00335, 2021.

[48] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, R. Timofte, “SwinIR: Image Restoration Using Swin Transformer”. arXiv preprint arXiv:2108.10257, 2021.

[49] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. We, et al., “Swin Transformer: Hierarchical Vision Transformer using Shifted Windows”. ICCV 2021.

[50] Q. Cai, Y. Qian, J. Li, J. Lv, Y. Yang, F. Wu, D. Zhang, “HIPA: Hierarchical Patch Transformer for Single Image Super Resolution”. arXiv preprint arXiv:2203.10247, 2022

[51] M. Conde, U. Choi, M. Burchi, R. Timofte, “Swin2SR: SwinV2 Transformer for Compressed Image Super-Resolution and Restoration”. arXiv preprint arxiv.2209.11345 arXiv, 2022.

[52] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, et al., “Swin Transformer V2: Scaling Up Capacity and Resolution”. CVPR 2022.

[53] D. Rezende, S. Mohamed, “Variational inference with normalizing flows”. Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, pp. 1530–1538, 2015.

[54] A. Lugmayr, M. Danelljan, L. Van Gool, R. Timofte, “SRFLOW: Learning the super-resolution space with normalizing flow”. In ECCV, pages 715–732. Springer, 2020.

[54] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, S. Ganguli, “Deep unsupervised learning using nonequilibrium thermodynamics”. arXiv preprint arXiv:1503.03585, 2015.

[55] C. Saharia, J. Ho, W. Chan, T. Salimans, D. Fleet, M. Norouzi. “SR3: Image super-resolution via iterative refinement”. arXiv preprint arXiv:2104.07636, 2021.

[56] H. Li, Y. Yang, M. Chang, H. Feng, Z. Xu, Q. Li, Y. Chen. “SRDiff: Single Image Super-Resolution with Diffusion Probabilistic Models”. arXiv preprint arXiv:2104.14951, 2021.

[57] Z. Kong, W. Ping, J. Huang, K. Zhao, B. Catanzaro, “Diffwave: A versatile diffusion model for audio synthesis”. arXiv preprint arXiv:2009.09761, 2020.

[58] J. Ho, A. Jain, P. Abbeel, “Denoising diffusion probabilistic models”. NeurIPS, 2020.

[59] Z. Wang, A. Bovik, H. Sheikh, E. Simoncelli. “Image quality assessment: from error visibility to structural similarity”. IEEE transactions on image processing, 13(4):600–612, 2004.

[60] R. Zhang, P. Isola, A. Efros, E. Shechtman, O. Wang. “The unreasonable effectiveness of deep features as a perceptual metric”. CVPR, 2018

[61] https://blogs.nvidia.com/blog/2023/02/28/rtx-video-super-resolution/

[62] https://blogs.windows.com/msedgedev/2023/03/08/video-super-resolution-in-microsoft-edge/

[63] X. Wang, L. Xie, C. Dong, Y. Shan. “Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data”. In ICCVW, 2021.

[64] S. Bell-Kligler, A. Shocher, M. Irani. “Blind super-resolution kernel estimation using an internal-gan”. In NeurIPS, 2019.

[65] L. Wang, Y. Wang, X. Dong, Q. Xu, J. Yang, W. An, Y. Guo. “Unsupervised degradation representation learning for blind superresolution”. In CVPR, 2021

[66] A. Kappeler, S. Yoo, Q. Dai, A. Katsaggelos, “VSRNet: Video Super-Resolution With Convolutional Neural Networks”. IEEE Transactions on Computational Imaging, vol. 2, no. 2, 2016.

[67] J. Caballero, C. Ledig, A. Aitken, A. Acosta, J. Totz, Z. Wang, W. Shi, “Real-Time Video Super-Resolution with Spatio-Temporal Networks and Motion Compensation”. arXiv preprint arXiv:1611.05250, 2016.

[68] X. Wang, K. Chan, K. Yu, C. Dong, C. Loy, “EDVR: Video Restoration with Enhanced Deformable Convolutional Networks”. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2019.

[69] M. Chu, Y. Xie, J. Mayer, L. Leal-Taixe, N. Thuerey, “Learning temporal coherence via self-supervision for GAN-based video generation”. In ACM Transactions on Graphics, vol.38, 2020. doi 10.1145/3386569.3392457.

[70] C. Liu, H. Yang, J. Fu, X. Qian, “Learning Trajectory-Aware Transformer for Video Super-Resolution”. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5687-5696, 2022.

[71] Z. Geng, L. Liang, T. Ding, I. Zharkov,” RSTT: Real-Time Spatial Temporal Transformer for Space-Time Video Super-Resolution”. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 17441-17451, 2022.