Literature Review: Edge Adaptive Video Upscaling

Abstract— **This is a review of image-scaling algorithms throughout the last several decades. Our focus will be on algorithms which can be practically implemented for video scaling however we will touch on the entire field of image scaling. Furthermore, we will be focusing on super-resolution, that is to say increasing the resolution of the resulting scaled image. Our review will be separated in the following parts:**

1) **Basic Interpolation**
2) **Frequency Domain Methods**
3) **Back-Projection and Optical Flow Methods**
4) **Adaptation Based on Local Information**
5) **Neural Networks**

I. Introduction

Image-scaling has been studied for decades and is still an active area of research. As a subsection of computer vision, it has been the object of study from many different angles over the years. Image-scaling with the goal of improving the resolution of the scaled image is called super-resolution. The problem of super-resolution is arguably ill-posed. Firstly, due to the very nature of the problem: it is a problem of many to one mapping ie. multiple HR (high resolution) images can correspond to the same LR (low resolution) image. Further, quantitative evaluation of proposed methods often does not match qualitative evaluation.

Super-resolution (SR) can be considered from several different perspectives: single image SR or SISR and multi-image SR or MISR. The former of these involves increasing the image resolution by predicting missing pixels through methods such as interpolation and statistical analysis. The latter involves using multiple displaced images, or frames, of the same scene and fusing them in order to recover lost information. Another perspective would be from the frequency domain, where an algorithm attempts to recover high-frequency components most commonly by extending the frequency spectrum and the spatial domain, where the algorithm attempts to recover the high resolution (HR) image from relationships between pixels or from inference.

the final, common way to break the problem down is into reconstruction based methods and learning based methods. The former of the two attempts to recover high frequency components of an image from local information or from extracting information from several successive frames. The latter, increases resolution by inferring the high frequency components from a large data-set.

II. Basic Interpolation

The earliest methods of super-resolution are based around interpolation and can mostly be classifies as SISR, reconstruction, spatial domain methods. These methods are still used today and due to their simplicity and computational efficiency. However, these methods have drawbacks in terms of the quality of the produced HR image.

a) Bilinear: Arguably the first and most basic methods were bilinear and nearest neighbour interpolation. Bilinear interpolation simply assumes a linear relationship between the intensities of the pixels in a given neighbourhood. [1] The LR image pixels need to be mapped onto an HR map corresponding to their relative positions in the LR image so that accurate interpolation can be carried out; This process will be true of most SISR super-resolution techniques. The process of interpolating occurs in two orthogonal directions. Note that often, before interpolation a Gaussian filter would usually be applied to smooth the image and remove noise. However, because of the linear assumption beforehand, the linear interpolation may not accurately capture complex or non-linear relationships between data points. [2]

b) Bi-cubic: Similar to bilinear, bicubic interpolation operates cubic interpolation twice, which uses cubic polynomials to interpolate between data points. A cubic polynomial is in the form of $f(x) = ax^3 + bx^2 + cx + d$. And the goal in

cubic interpolation is to find the four coefficients of the cubic polynomial that best fits the given data points. These coefficients are determined using the values of the neighboring data points and at least four data points.[3]

The general form of a bicubic interpolation equation for a target pixel at position (x, y) is:

$f(x, y) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij} \cdot x^i \cdot y^j$

where a(i, j) are the coefficients to be determined. Bicubic interpolation involves more complex calculations and has higher computational complexity compared with linear way. On the other hand, because of the complex calculations, bicubic interpolation is able to capture more intricate patterns and structures and less block artifacts. These make the image after bicubic interpolation have a higher image quality.

c) Lanczos: Lanczos resampling, named after its originator Cornelius Lanczos, is a sophisticated, windowed sinc-function-based interpolation method commonly employed in digital image processing. It provides superior performance in retaining high-frequency details during the resampling process compared to other standard methods such as bilinear or bicubic interpolation.

The essence of the Lanczos method lies in the application of a sinc filter, coupled with a finite impulse response (FIR) filter, generally referred to as a window function. The sinc function, derived from the Fourier transform of a rectangular function, is ideal for bandlimiting a signal to avoid aliasing. However, due to its infinite extent, it is not practical for direct implementation. Consequently, a window function is introduced to confine the sinc function within a finite extent, resulting in the Lanczos kernel. Lanczos interpolation can be described by the equations below[4]:

$$L(x) = \begin{cases} \text{sinc}(x)\text{sinc}(x/a) & \text{if } -a < x < a \\ 0 & \text{otherwise} \end{cases}$$

Equivalently,

$$L(x) = \begin{cases} 1 & \text{if } x = 0 \\ \frac{a \sin(\pi x) \sin(\pi x/a)}{\pi^2 \pi^2} & \text{if } 0 < |x| < a \\ 0 & \text{otherwise} \end{cases}$$

A distinct advantage of the Lanczos resampling method is its ability to reproduce high-frequency details more accurately compared to other interpolation techniques. This is due to the better frequency response of the Lanczos kernel, particularly in the stopband region, which minimizes the aliasing effects. This characteristic makes it an attractive option in image processing applications where preserving fine details during resampling is a priority.

The parameter a is a positive integer, typically 2 or 3, which determines the size of the kernel. We determine the weight L(x) corresponding to different positions in the window based on the input point, and then take the weighted average of the point values in the template[5], follows equation like this,

$S(x) = \sum_{i=x-a+1}^{x+a} s_i L(s - i)$

Overall, Lanczos interpolation is widely used due to its fast speed, good result with rather low cost.

III. Frequency Domain methods

Analysis of images in the frequency domain has given the basis for much of the research surrounding super-resolution by outlining the fundamental mathematical principles surrounding it. In this domain, one considers the optical system to have a transfer function which attenuates frequencies beyond a certain cut-off [6]. Therefore, in order to reconstruct the image, one must extend this spectrum. The main principle which allows SISR in the frequency domain is that "a function of a complex variable is determined throughout the entire Z-plane from a knowledge of its properties within an arbitrarily small region of analyticity."[7] This means that the information needed to extend the spectrum (for a finitely sized object) is available in any analytical sample of the frequency spectrum of the function. The proof of analyticity and in-depth analysis of frequency spectrum extension is available in [6].

A. Fourier Based Methods

a) : The method mentioned here simply outlines the mathematical principles and implements a proof of concept rather than an applicable SR algorithm. This paper proves that the limit to restoration is imposed not by how precise the optical system is but by the noise introduced in the system. Another approach [8], based on the above

principles, considers a ”segment of the known spectrum to be the sum of the complete spectrum plus an error spectrum which is equal and opposite to the true spectrum outside the given segment.” In this way, the authors are able to devise an iterative algorithm to minimise the error spectrum energy.

b) : The above methods both propose extending the spectrum of the image and are both SISR methods, the first MISR method was presented in [9]. This method was developed for improving images from satellites which are globally shifted which allows the researchers to use the Fourier transform's shifting property [10] to take into account the shifts. However this method illustrates the flaw in the Fourier approach to super-resolution; though it provides the theoretical basis of SR, this method, as well as those mentioned above, require small, global motion and suffer greatly if the motion estimation is not correct as stated in [9]. For our purposes of video-scaling, where motion is local and complex, Fourier methods are not applicable. We touch on modern methods of motion estimation and MISR in section 3.

B. Wavelet Based Methods

Wavelet-based methods offer more practicality compared to Fourier-based methods. Instead of merely providing a frequency domain representation of the signal, wavelet transforms represent the signal in both time and frequency domains, which enables access to localized information about the signal. In general, wavelet-based methods use the low-resolution frame to replace the low-frequency wavelet-transformed sub-band[11], then adjust the high-frequency sub-bands with different approaches. All the sub-bands are subsequently fused and inverse wavelet-transformed to produce the high-resolution frame.

1) Direct Wavelet Transform Based Methods: This series of methods utilize direct wavelet transform (DWT) to produce four sub-bands from the original video frame, with one low-frequency sub-band consisting of LL sub-band and three high-frequency sub-bands consisting of LH, HL and HH sub-bands. The LL sub-band is replaced by the original low-resolution frame, optionally up-scaled by bicubic interpolation; whereas the high-frequency sub-bands are generally estimated or refined using motion estimation based on the current and nearby frames and then interpolated using bicubic interpolation. The resultant sub-bands produce the high-resolution video frame through IDWT.

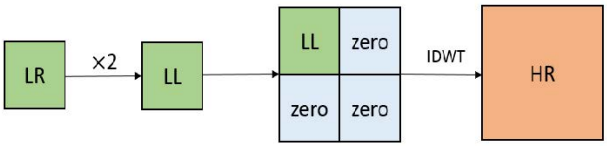

Fig. 1. Wavelet Zero Padding [12]

a) Wavelet Zero Padding: Wavelet zero padding is the simplest discrete wavelet transform based method. As shown in figure 1, it solely takes in every single low-resolution frame as the low-frequency LL sub-band, after multiplying all its coefficients by 2 and up-scaling it with bicubic interpolation; fills the high-frequency sub-band coefficients with zeros, then performs inverse discrete wavelets transform (IDWT) to reconstruct the single high-resolution frame [12]. It is simple and fast, but the super-resolved frame produced does not have any edge enhancement and appears blurry. Besides, it can only reliably upscale the video frame by 2; other scaling factors are possible through interpolating the LL sub-band, but the performance is not ideal and primarily dependent upon the interpolation method.

b) Image Registration with Block Matching: Image Registration can ideally yield correct high-frequency sub-band coefficients and high-resolution video frames, if it is done accurately. Block matching is one of the least computationally demanding image registration methods. Multi-frame registration is a more intuitive approach, yet has low accuracy if the original frame has large motion blur or occluded objects.

To acquire high-accuracy image registration, multi-scale registration, which utilizes self-similarity within a single frame, is proposed in [13]. As shown in figure 2, the video frame is first DWTed to multi-scale sub-bands of high and low frequency. In the wavelet domain, using exhaustive search block matching, based on an evaluation function of the sum of squared difference values, overlapping regions of pixels in the original video

frame Y^0 are aligned to the down-scaled low-frequency sub-bands $\{Y^n_{LL}|n = 1,2,3,4,5,6\}$. Noticeably, the block size is the same for all the scaled sub-bands, meaning that similar objects of various sizes will all be matched together. Using the alignment results, for the matched blocks, the down-scaled high-frequency sub-band $\{Y^n_{LH}, Y^n_{HL}, Y^n_{HH}|n = 1,2,3,4,5,6\}$ coefficients are assigned with a window function to the corresponding block within the high-frequency sub-band of the same size as the original frame $\{Y^0_{LH}, Y^0_{HL}, Y^0_{HH}\}$. If after expanding the block by 2, the difference between coefficients of the expanded block from the low-frequency sub-band Y^{n-1}_{LL} and the original frame Y_0 is less than a threshold, the coefficients from corresponding blocks in $\{Y^{n-1}_{LH}, Y^{n-1}_{HL}, Y^{n-1}_{HH}\}$ are assigned to the block in the up-scaled high-frequency sub-bands $\{Y^{-1}_{LH}, Y^{-1}_{HL}, Y^{-1}_{HH}\}$. A single super-resolved video frame is then produced using the original video frame Y_0, two levels of high-frequency sub-bands $\{Y^0_{LH}, Y^0_{HL}, Y^0_{HH}\}$ $\{Y^{-1}_{LH}, Y^{-1}_{HL}, Y^{-1}_{HH}\}$ through two-level inverse wavelet transform.

This method produces videos with noticeably sharp edges and without unpleasing artifacts or blurs, while yielding a very high PSNR and SSIM. Since the threshold ensures that the high-frequency sub-bands of the super-resolved video are similar enough to the original video. The adoption of step-search-like block matching for registration reduces the computational cost and is advantageous for real-time processing. This method is proposed only for upscaling with a factor of 4, however, upscaling with other even numbers based on it is possible with minor modifications, which could further investigated.

c) Weighted Motion Compensation: In order to accomplish image registration and motion estimation more accurately, it is proposed to utilize both spatial and temporal information in the wavelet domain, through the aid of weighted motion compensation (WMC) in [14]. The low-resolution video frames are firstly Lanczos interpolated and DWTed. Then WMC is performed in the wavelet domain. In other words, for every block of pixels in one frame, the four frequency sub-bands from DWT each produce a motion vector and a predicted block for the next frame. The motion

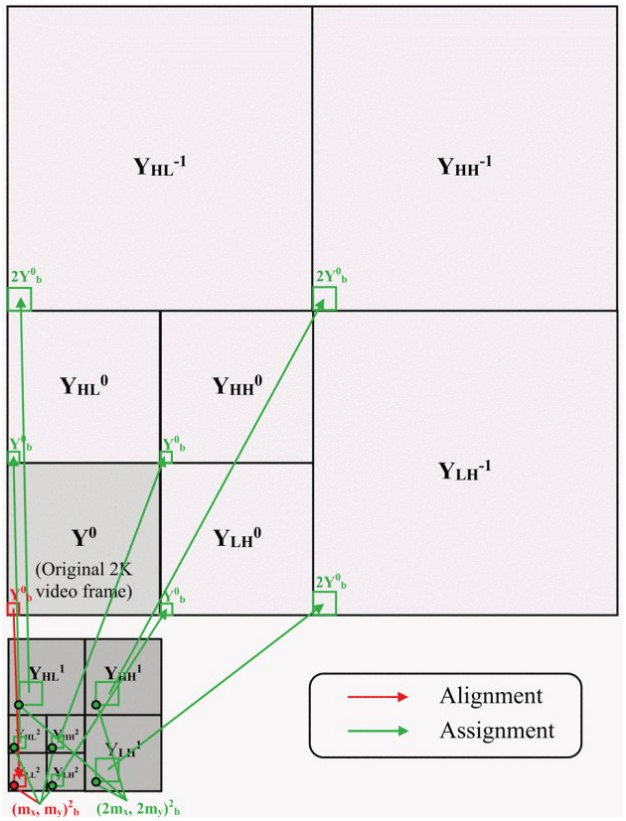

Fig. 2. Alignment and Assignment Process for Block Matching Image Registration [13]

vectors can be estimated using the block-matching method discussed before, which can also be encoded in compressed videos to save computational costs. The final predicted block in each sub-band in the next frame is a weighted sum of predicted blocks from all the frequency sub-bands in the previous frame, which is given by

$$B^{sb}_W = w_1 \times B^{sb}_{mvLL} + w_2 \times B^{sb}_{mvLH} + w_3 \times B^{sb}_{mvHL} + w_4 \times B^{sb}_{mvHH} \quad (1)$$

where sb is any sub-band from LL, LH, HL, HH; B^{sb}_{mvLL}, B^{sb}_{mvLH}, B^{sb}_{mvHL}, B^{sb}_{mvHH} are blocks predicted by using $MV^{B_w}_{LL}$, $MV^{B_w}_{LH}$, $MV^{B_w}_{HL}$, $MV^{B_w}_{HH}$, which are motion vectors obtained by motion estimation in the four sub-bands individually. The process is shown in figure 3. The use of WMC greatly enhances the accuracy of estimating complex motions, which are crucial for producing correct super-resolved videos.

To further enhance the edges, wavelet difference coefficients (WDC) are estimated to yield higher

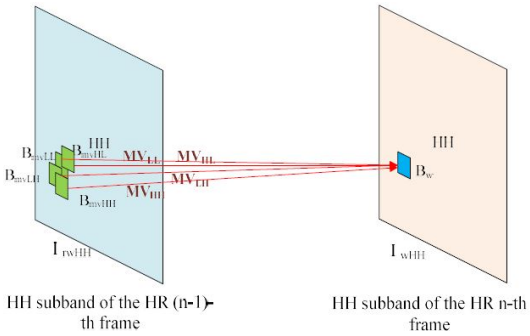

Fig. 3. Weighted Motion Compensation [14]

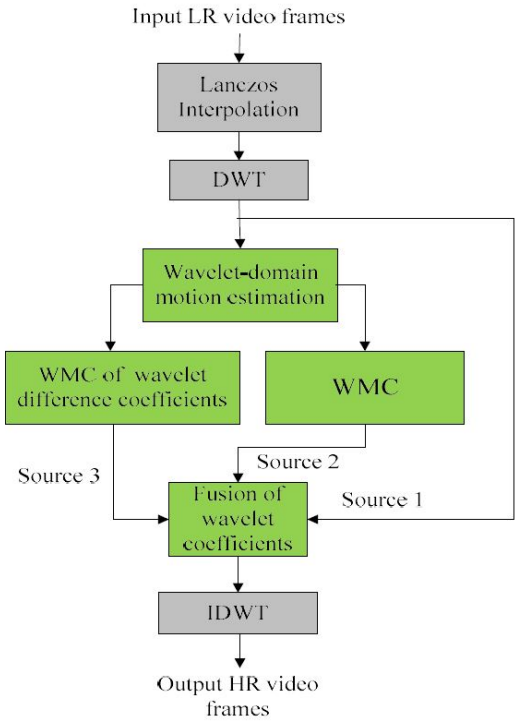

Fig. 4. Overall WMC DWT Algorithm Structure [14]

frequency components. WDC is the difference of wavelet coefficients between the $n-1$ super-resolved frame and the $n-1$ Lanczos-interpolated frame, which represents the information about the edges missing from the Lanczos-interpolated frame but existing in the detailed super-resolved frame. The $n-1$ super-resolved frame is obtained recursively, except for the first frame which is obtained through bicubic interpolation. The WDCs are processed by WMC and then added to the Lanczos-interpolated DWT coefficients.

The Lanczos-interpolated DWT coefficients, WMC and the Lanczos-interpolated DWT coefficients modified based on WDC, are combined linearly to produce the final super-resolved frame. The weight of each term is the normalized reciprocal of the Euclidean norm between the IDWT of them and the original low-resolution frame.

This method, which is shown in figure 4, produces videos with more sharp edges and coherent frames, and has a relatively low computational cost compared to other motion compensation methods for improving the motion estimation. The usage of adaptive weights also ensures that the final high-resolution video frame is as close as possible to the original low-resolution frame. Nonetheless, it is experimentally shown that this method, or motion estimation based method, is limited to videos containing local or object motions. For videos containing dynamic and complex motions, the PSNR gain of this method is small compared to the more traditional ones including Lanczos and bicubic interpolation; although for more static videos, the gain is distinctive.

This method has only been experimented for a scaling factor of 2, however, any other scaling factors are possible as long as the interpolation method and motion estimation algorithm used permit them.

d) Image Registration with Non-Local Means: To bypass motion estimation, which often yields inaccurate motion vectors and final super-resolved results when complex motion changes between frames are involved, Non-Local Means (NLM) based method is used in [15]. With the assumption that a small block of pixels appears several times in one frame and also in different frames, wavelet domain coefficients corresponding to a pixel can be estimated using a normalized weighted sum of wavelet domain coefficients corresponding to all pixels in current and nearby frames, in other words, by using non-local means. The estimate of a pixel at position (i,j) is given by

$$\hat{z} = \frac{\sum_{t=1}^{T} \sum_{(k,l)\in N(i,j)} \omega_t(k,l) y_t(k,l)}{\sum_{t=1}^{T} \sum_{(k,l)\in N(i,j)} \omega_t(k,l)} \quad (2)$$

where y are the original low resolution wavelet coefficients, t is the frame index and w are the filter weights, which are based on the similarity

between pixel blocks and given by

$$w(k,l) = \exp\{-\frac{\left\|p_{y(i,j)} - p_{y(k,l)}\right\|_2^2}{2\sigma^2}\} \cdot f(\sqrt{(i-k)^2 + (j-l)^2 + (t-1)^2}) \quad (3)$$

The original frame is firstly DWTed and Lanczos interpolated, then coefficients in HL and LH sub-bands are interpolated using an arbitrary wavelet interpolation method and re-estimated through NLM, unless the coefficients have large magnitudes satisfying $|M_{coef}| > \alpha \times std(|M_{coef}|)$; the interpolated original frame forms the LL sub-band and HH sub-band is filled with zeros, as shown in figure 5.

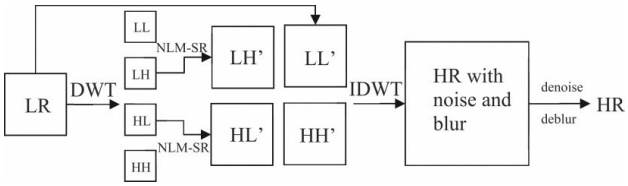

Fig. 5. Overall DWT with NLM Algorithm Structure [15]

This method avoids explicit motion estimation; instead of estimating a target pixel position in nearby frames, it simply considers all possible positions where the pixel may appear and provides a better estimate of the coefficients when motion changes between frames. This method yields sharper edges and fewer errors around the edges, outperforms former methods including NLM-SR and single-frame wavelet interpolation both in terms of PSNR and visual quality. However, it needs extra denoising and deblurring filters, although it is still a relatively fast method, particularly if fast denoising filters are used; this method also does not consider HH sub-band, which introduces noise and blurs.

This method has only been tested for a scaling factor of 2, however, any other scaling factors are theoretically possible as long as the interpolation method and the denoising and deblurring filters used permit them.

e) Enhanced with Stationary Wavelet Decomposition: The redundancy and shift-invariance of the DWT, which means that DWT coefficients are inherently interpolable, makes them suitable for preserving high-frequency components in video super-resolution. Nonetheless, in DWT, the sub-bands are all down-sampled from the original video frame, leading to information loss; stationary wavelet transform (SWT), which is similar to DWT but produces sub-bands with the same size as the original video frame and preserves more high-frequency components, can be used to modify the high-frequency sub-bands of DWT, thereby reduces the information loss and produce super-resolved videos with sharper edges, as proposed in [11].

The low resolution video frame is first SWT and DWT. The DWT high-frequency sub-bands are first bicubic-interpolated and then incremented by the corresponding high-frequency coefficients from SWT. The modified DWT results and original video frame are bicubic-interpolated to be used respectively as the high and low-frequency sub-bands for IDWT to create the super-resolved video frame, as shown in figure 6.

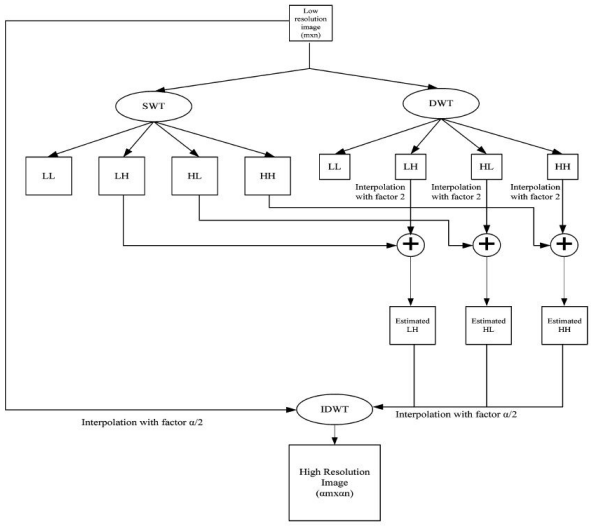

Fig. 6. Overall DWT with SWT Algorithm Structure [11]

This method utilizes only simple bicubic interpolation of wavelet transformed coefficients, without any forms of motion estimation, which greatly reduces the computational cost, since wavelet transform is relatively fast. On the other hand, it is effective at enhancing edges, through the addition

of SWT coefficients, providing superior PSNR and visual results compared to simple wavelet transform based methods.

This method only works with even-number upscaling factors, since it involves upscaling all sub-bands by $\frac{overall upscaling factor}{2}$. Other scaling factors may be possible if a interpolation algorithm for non-integer factors is used instead, such as the method proposed in [16].

f) Enhanced with Edge Directional Interpolation: Edge directional interpolation (EDI) classifies pixels as edge and non-edge pixels and applies different interpolation algorithms to preserve the edge structure. It can be used to firstly produce an up-sampled version of the low-resolution input frame, which is then DWTed to yield high-frequency sub-bands. The high-frequency sub-bands are then combined with the original input frame as the low-frequency sub-band to produce the final super-resolved frame, through IDWT, as shown in figure 7. Although this method proposed in [12] is mostly a switch of the order of DWT and interpolation, compared to the conventional DWT-based methods, it produces visually sharper frames, which are particularly noticeable when many edges and textures are present.

This method is tested for a scaling factor of 2. Other integer scaling factors are possible if the sub-bands are interpolated in the process; non-integer scaling factors should also be possible depending on the interpolation method used.

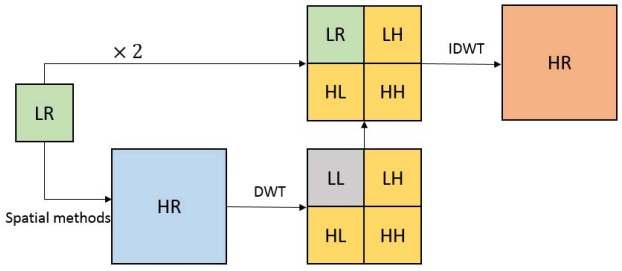

Fig. 7. Overall Algorithm Structure [12]

2) Complex Wavelet Transform Based Methods: This series of methods utilize dual-tree complex wavelet transform (DT-CWT) of the low-resolution video frame to produce one low-frequency sub-band consisting of LL sub-band and six high-frequency sub-bands in different directions: $+75°$,$+45°$,$+15°$,$-15°$,$-45°$,$-75°$ [17]. The LL sub-band is replaced by the original low-resolution frame, with its all coefficients multiplied by 2 and up-scaled by bicubic interpolation; the high-frequency sub-bands are also interpolated. The resultant sub-bands produce the high-resolution video frame through inverse dual-tree complex wavelet transform(IDT-CWT), as shown in figure 8.

Similar to DWT, DT-CWT is shift-invariant, meaning that the resultant coefficients are inherently interpolable and can be used readily for super-resolution. Different from DWT, DT-CWT can decompose frames in different high-frequency directional sub-bands, which isolates the edge details in different directions and reduces the inter-directional interference in the super-resolution process, resulting in more distinctive edges.

In theory, DT-CWT based methods can upscale video frames with any arbitrary real-number scaling factors. To upscale the final frame by α, the high-frequency sub-bands need to be interpolated with a factor of α and the original frame needs to be interpolated with $\frac{\alpha}{2}$. Nonetheless, the specific interpolation algorithm limits the range of possible α, typically to integers.

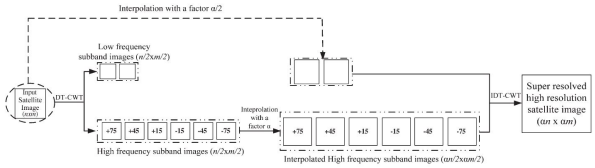

Fig. 8. Typical Structure for DT-CWT Based Algorithms [17]

a) with Bicubic Interpolation: The relatively simple method for up-scaling high-frequency sub-bands is bicubic interpolation, as proposed in [17]. However, this results in a relatively poor performance in both PSNR and visual quality. Nonetheless, the computational cost for wavelet transform and bicubic interpolation is relatively low. It also does not require any motion estimations or denoising and deblurring filters. Most importantly, this method can already benefit from the main advantages of DT-CWT based methods, which are the preservation of high-frequency components demonstrated by edges after interpolation and reduction in inter-directional interference.

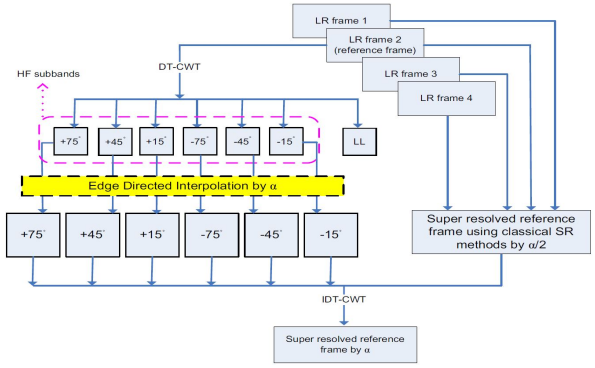

Fig. 9. Structure of Multi-Frame EDI DT-CWT Algorithm [18]

b) with Edge Directional Interpolation: Edge directional interpolation (EDI) classifies pixels as edge and non-edge pixels and applies different interpolation algorithms to them to preserve the edge structure. Interpolating high-frequency sub-bands using EDI, as proposed in [19], yields super-resolved videos with noticeably fine and sharp edge details. Additionally, to acquire a low-frequency sub-band that gives a super-resolved video with finer details, the nearby frames of the current frame being super-resolved can be taken, along with the current frame, for multi-frame super-resolution using an arbitrary method, which is proposed in [18] and shown in figure 9. This is particularly helpful for low-resolution video sequences, since the nearby frames can be viewed as multiple samples of the original frame.

This method yields super-resolved videos with sharp edges and considerably outperforms former DT-CWT methods in PSNR. However, this method has not yet been implemented in real-time, although that is a possibility.

IV. Optical Flow and Back-projection

Fundamentally, MISR can be summarised as mapping an image of the same scene, distorted in some way **[cite something]** (eg. translation) hereon out referred to as the distorted image onto a reference image; for the purpose of video-scaling this would involve mapping a subsequent frame onto the current one, and finally fusing the two. Back-projection and optical flow methods approach this very directly. One early example [20], inspired by the imaging technology employed in CAT scans involves making an initial guess for the HR image and the "back-projection" of the differences in subsequent images to improve the guess. The problem with the above approach is that it estimates global translation and would therefore not be applicable to video.

In order to perform local motion estimation across the entire image, one can employ optical flow, this method originally described in [21] has become a basis for motion estimation in computer vision. It is founded in three assumptions: motion between frames is small and that pixel intensities are constant. These two assumptions produce the optic flow constraint equation:

$$I_x u + I_y v + I_t = 0 \tag{4}$$

This can be rewritten as:

$$[I_x(p) + I_y(p)] \begin{bmatrix} u \\ v \end{bmatrix} = -I_t(p) \tag{5}$$

The above is an under-determined system, Lucas and Kanade resolve this by introducing the third and final assumption of spatial coherency which turns the above into an over-determined system of linear equations which can be solved via the Moore-Penrose inverse.

This feasibility of this method in super-resolution is discussed in [22]. This paper concludes that super-resolution with optical flow based methods is feasible, however some frequency-domain analysis suggests that in order to estimate high-frequencies the model needs to be very accurate, however, small gradients in the image present higher error in flow estimation. The method used in this paper is an augmented version of the method proposed in [23] which uses expensive image segmentation and object tracking techniques to register the image which here, has been replaced by the optical flow method while The super-resolution technique remains the same. This technique is a traditional correcting algorithm which uses N LR images to improve the estimate (of the HR pixel) iteratively, N times. The authors conclude that with small noise, and accurate motion estimation, optical flow is an effective method of super-resolution.

The above method uses dense flow estimation which is computationally expensive, sensitive to image noise and somewhat prone to error in motion estimation. A more modern approach in [24] seeks to perform local flow estimations only

at selected interest points which are determined, in this case by the SIFT interest point detector [25]. Interest point matching is far more robust than dense flow estimation due to the accuracy of the SIFT feature, a double check can also be performed between each feature ie. from the 'reference' frame to the 'distorted' frame and vice versa thereby making it extremely unlikely to have a false match.

In order to construct a support region ($\mathcal{R}$) for which to apply the calculated transformations between the interest points (note that SIFT features are invariant to scale and rotation thus allowing affine transformations (3) to represent motion), a confidence map, utilised in [26] and described in (4) is combined with Canny edge-detection [27].

$$T_i = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

$$\mathcal{R} = |B_k(L^r(T_i\vec{x}) - L^d(\vec{x}))| < \eta_{map} \tag{7}$$

Where B_k is the blurring operator, L^r is the reference frame and L^d is the distorted frame. If the difference between the transformed image and the distorted image is small, then the corresponding pixel location in the confidence map is set to 1. Further, for a pixel to be taken in the support region, both: the pixel and its nearest edge point must fall within the confidence map. The neighbourhood of each interest point is expanded until a connected path exists around each support region. If there exists overlap, pixels are assigned to the region of the nearest interest point.

The problem achieving MISR through mapping frames onto one another is a difficult one in real world scenarios due to noise. However it can be simplified by focusing on specific regions. Drawbacks in terms of computational complexity are prevalent however, and the above algorithms would need to be simplified significantly in order to be applicable to real-world scenarios.

V. Adaption Based on Local Information

Often, algorithms which are optimised for computational efficiency are SISR, reconstruction algorithms which adapt interpolation coefficients based on local structure. An early and incredibly simple algorithm is proposed in [28]. This algorithm is best understood in 1D then generalised to 2D.

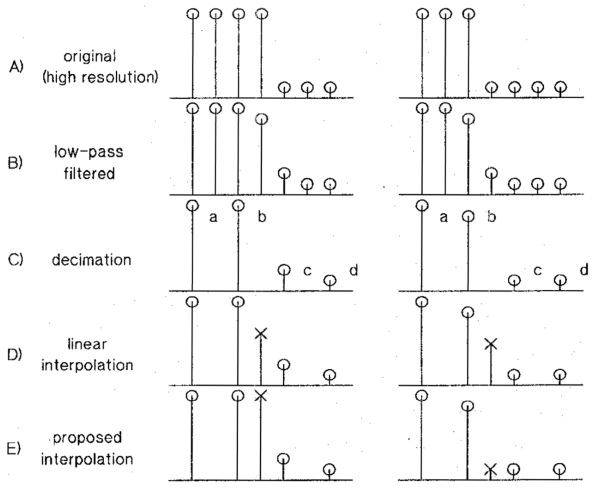

Fig. 10. From [28], the two columns show the two different possibilities

a) : Figure one shows the process from image capture, which is characterised by a low-pass filter effect and decimation to the interpolation. The proposed interpolation method is defined as follows:

$$x = \mu b + (1 - \mu)c \tag{8}$$

$$\mu = \frac{k(c-d)^2 + 1}{k((a-b)^2 + (c-d)^2) + 2} \tag{9}$$

Where k is a user-selected parameter which determines the edge sensitivity. When the edge is midway between b and c, a-b = c-d, so that μ = 0.5 and x = (b+c)/2. Under this condition, the filter behaves as a linear interpolator. However, when the edge is asymmetrically located, eg. when a - b $<$ c - d, so that $\mu > 0.5$, x is now $\approx$ b. The above can easily be extrapolated to two dimensions where the interpolation is simply performed along the horizontal and vertical axes.

This method is incredibly simple, however, it forms the basis of this family of methods which have become more complex and far better performing. A famous and often cited approach is the NEDI (new edge-directed interpolation) [29]. This approach is based on a statistical approach enabled by the geometric duality between the LR and HR co-variances. Geometric duality refers to

the correspondance between local covariances of an LR and HR pixel which are along the same orientation (relative to an edge).

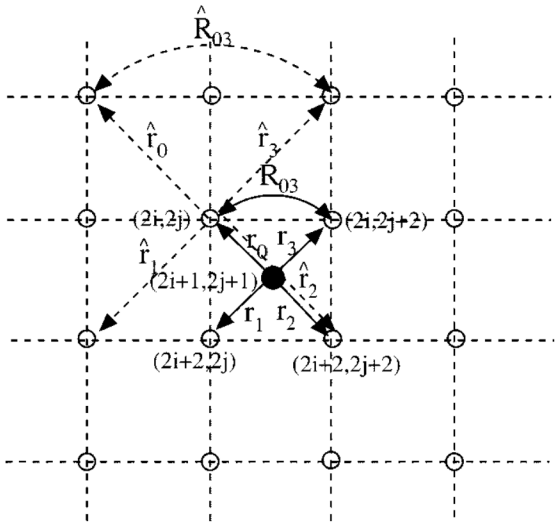

Fig. 11. illustration of geometric duality: R is the covariance

The approach is a continuation of a previous work which performs least-square based adaptive prediction for lossless compression; the details of which are available in [30]. In this paper is addressed the edge-directedness property in detail, however, for completeness, it can be summarised as such: A double rectangular window (fig. 3) is used to obtain prediction neighbours in matrix C in (7).

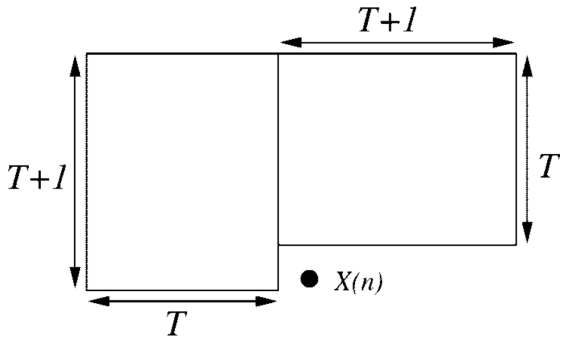

Fig. 12. "training" window used to optimise prediction coefficients

$$C = \begin{bmatrix} X(n-1-1) & \dots & X(n-1-N) \\ \vdots & & \vdots \\ X(n-M-1) & \dots & x(n-M-N \end{bmatrix} \tag{10}$$

It is then straight forward to see that when the matrix C contains edge neighbours, it is often full rank. The authors implement a hybrid method between covariance-based adaptive interpolation and linear interpolation to greatly reduce computational complexity.

Despite the efforts to make the algorithm efficient, auto-regression methods will always suffer from long computation times due to the need to solve sets of linear equations. An alternative method, proposed in [31] suggests obtaining HR pixels via a weighted sum resulting from some arbitrary interpolation in two directions, where the weights are chosen based on the error of this arbitrary model when applied to existing LR pixels:

1) apply arbitrary interpolation along the $45°$ and $135°$ diagonals for the four (LR) nearest neighbours of the missing (HR) pixel as exemplified in fig. 4.
2) calculate the difference in values obtained from the interpolation and the true pixel values for the neighbours.
3) sum the errors for each diagonal for the four neighbouring pixels such that two error values: one for the $45°$ and one for the $135°$ diagonals are obtained.
4) weight the obtained values (in step 1) for the missing HR pixel according to the weights, (8) shows the weighting used by the authors.

$$W_{45} = \frac{Err_{45}^{sf}}{Err_{45}^{sf} + Err_{135}^{sf}} \tag{11}$$

Where sf is a scaling factor obtained experimentally in order to make the algorithm more sensitive to the edges. W_{135} is obtained by 1-W_{45}.

This algorithm, in some sense, implements the covariance properties ina much more efficient way. The authors also propose a pipe-lined VLSI implementation which is able to support the up-scaling to UHD at 30fps with clock frequency of 297MHz.

There are other methods like importing sobel filter. In that situation, the edge of the original image is detected by using four Sobel spatial operators, which is proposed in [32] and shown in figure 13. And the direction of edge is then measured by using two newly proposed edge detectors employing two experimentally determined threshold

values. Based on these, pixels are interpolated adaptively, where non-edge pixels are interpolated through bilinear interpolation and edge pixels are interpolated according to the edge direction:

1) if edge is in $0°$ direction, pixel a is assigned with value $\frac{I+II}{2}$
2) if edge is in $90°$ direction, pixel c is assigned with value $\frac{I+IV}{2}$
3) if edge is in $45°$ direction, pixel b is assigned with value $\frac{II+IV}{2}$
4) if edge is in $135°$ direction, pixel b is assigned with value $\frac{I+III}{2}$

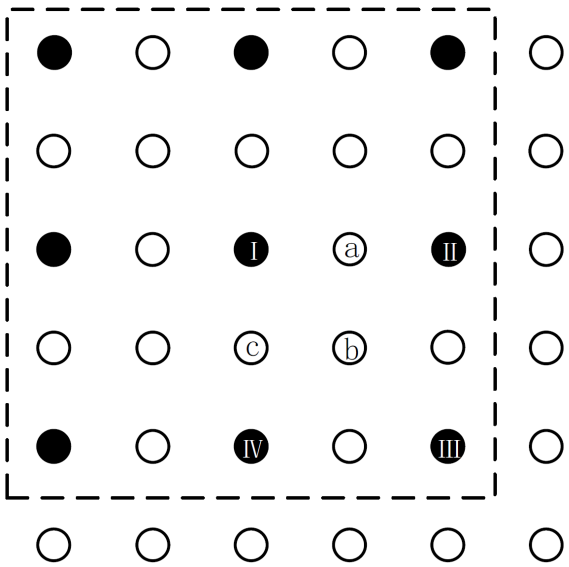

Fig. 13. Sobel based Edge Interpolation [32]

This algorithm can achieve edge adaptive interpolation, with very little computational cost compared to NEDI. The resultant upscaled video frames have relatively sharp edges, but also with zig-zag artefacts on the edges.

To suppress the zig-zag artefacts, the algorithm is improved in [33]. The improved algorithm uses canny edge detector to create an edge map; the sobel operators are then applied to the edge map to evaluate the edge direction, using Sobel gradients and threshold values obtained from the mean and standard deviation of Sobel gradients. The edge pixels are still interpolated using the same adaptive method proposed in [32]. Pixels next to every detected edge pixels, which lie along the edge pixel's edge direction, are also interpolated adaptively as a weighted sum of its original value and the nearest pixel in the same direction, as shown in figure 14. For instance, if the edge pixel is in $45°$ direction, $f(x-1,y-1)$ is assign with $W_{45}\dot{f}(x-1,y-1)+(1-W_{45})\dot{f}(x-2,y-2)$, in which $W_{45}=\frac{f(x-1,y-1)}{f(x-1,y-1)+f(x-2,y-2)+1}$.

This algorithm can effectively smooth the zig-zag artefacts along the edges produced by adaptive interpolation in [32]. It also achieves this with more complicated adaptive interpolation, but maintains approximately the same low computational cost as the former algorithm through only using Sobel operators on the Canny edge map. Nonetheless, the resultant video frames may have less sharp edges.

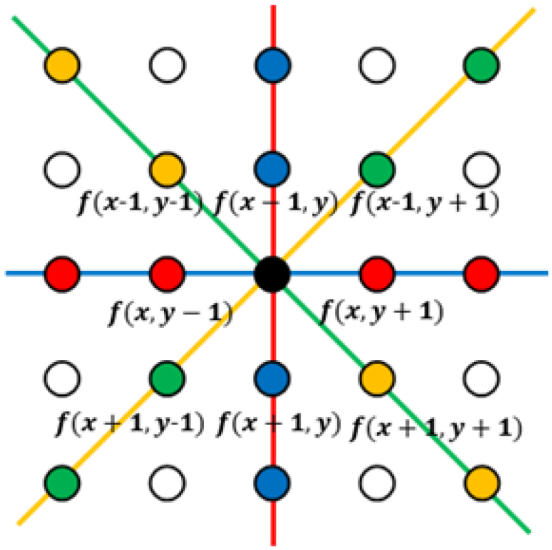

Fig. 14. Zig Zag Suppression (Axis Rotated) [33]

VI. Neural Network

A. Neural Network performance with software

Convolutional neural networks(CNN) have always been popular and effective in the field of image processing. The basic structure of CNN includes multiple interconnected layers to perform operations like convolution, pooling and activation. The convolutional layers with learnable filter can capture the local feature and spatial relationship of the entire images.

According to the comparison gallery[34], it it obvious that deep learning algorithm performs the best. However, deploying CNNs on Field-Programmable Gate Array(FPGA) for real-time

processing brings both advantages and challenges. The following review aims to analyze the pros and cons of using CNN and propose potential solutions to the address the challenges.

B. Parallel threads and GPU utilization

Using CNNs includes intensive matrix operation, leading to high computational complexity. Research about accelerating the video super resolution on FPGA[35] has shown that 97% of the time is spent on the convolutional computation, and it takes 450s to process a single frame. The article proposes a solution to this problem: applying efficient GPU optimization techniques and memory hierarchy of GPU to parallelize the convolution operation. The convolution operation is defined as an array operation where each output element is the weighted sum of the product of filter values with the corresponding input patch values. The computation at each pixel of the entire image is considered as a thread, as Figure15 shown, therefore the overall time of convolutional operation will be divided by the number of the parallel threads. Theoretically, the entire operation can be accelerated to the operation time equivalent to one pixel, but it will then require more than 700,000 threads which will cost too much resource in real projects. In the article [35], the method achieved a speed of 225X by performing parallel implementation of reasonable count of thread.

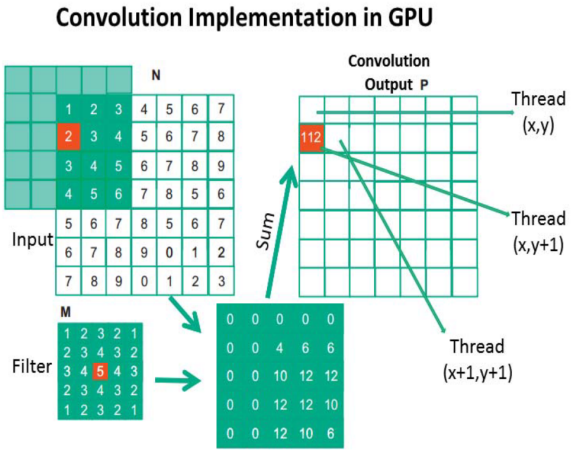

Fig. 15. Implementation of thread for convolutional operation

Another optimisation to consider is to do with memory management. The hierarchical GPU memory model consists of global memory, constant memory, shared memory and texture memory. Here shared and constant memory has fastest access time and global memory has slowest access time. In the previous method, the input parameters are pre-loaded into global memory. Each thread will access global memory to fetch the parameters and perform the computation. The unnecessary access to memory between convolution and activation can also be cancelled to speed the process faster.

C. Reducing the computing complexity from the architecture

As the depth of a neural network increases, the performance will be better but also the complexity increase. Thus, finding the balance between complexity and performance is essential to the optimisation. First of all, the format of the input data will affect especially when running the methods on FPGA platform. It is well known that 32-bit floating-point (single precision) data is used in most of the deep learning platforms such as PyTorch [36] and TensorFlow [37]. Therefore converting floating-point data to fixed-point data with optimal bit depth is essential to meet the trade-off between quality and complexity with limited resource.

The common inspiration behind different method is selective patch and reduce the computation over the entire images[38, 39, 40, 41]. One way is to convert the RGB channels to YCbCr channel[42], and only Y channel is used as input for CNNs. The PNSR performance shows that the CNN trained with only Y channel is similar to the CNN trained with RGB channels[40].

To achieve the small use of convolutional filter parameters and line memories, the network incorporates (i) depth-wise separable convolutions, (ii) 1D horizontal convolutions, and (iii) residual connections, each of which is described in detail in the following subsections. As a result, the number of parameters used in proposed network is about 21 times smaller than that of SRCNN-Ex [40], about 4.5 times smaller than that of FSRCNN [43], and 1.56 times smaller than that of FSRCNN-s [43], while maintaining similar PSNR and SSIM performance compared to that of SRCNN-Ex [40]. (i) Depth-wise separable

convolutions(DSC)[44] has shown to achieve similar classification performance only with one-ninth of the number of parameters, compared to the cases with conventional non-separable convolutions. The experiment results show as tableI below[38]

Method	Bicubic	SRCNN[40]	SRCNN-Ex[40]	FSRCNN[43]	FSRCNN-s[43]
# of Params	-	8k	57k	12k	4k
Bits width	-	32-bit	32-bit	32-bit	32-bit

TABLE I

EXPERIMENTAL RESULTS

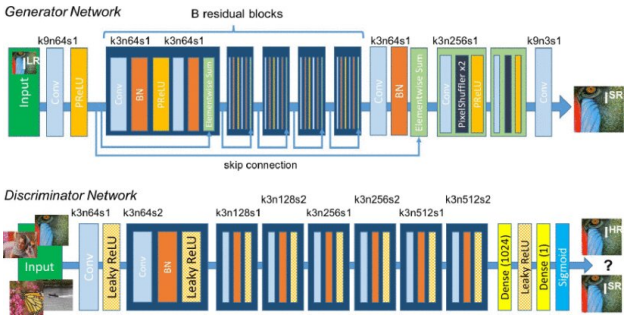

Fig. 16. The architecture flowchart of SR-GAN with corresponding kernel size (k), number of feature maps (n) and stride (s) indicated for each convolutional layer. [50]

D. The state-of-art architecture for Super-resolution

Generative Adversarial Networks (GANs) are powerful models in multiple image processing tasks, consisting of two key components: a generator and a discriminator, which are trained simultaneously in an adversarial manner[45]. The generator aims to produce the high-resolution images from the low-resolution images, while the discriminator attempts to distinguish between the generated HR-images and the real HR-images. The GANs model can be applied in lots of image processing tasks, such as text to image synthesis[46],image denoise[47, 48], image to image translation[49], image super-resolution[50] and etc. In the field of super-resolution, faster and deeper convolutional neural networks have already resulted well in both speed and accuracy[39, 40]. However, the issue of keeping high frequency information and containing the fine texture still exists. GANs have revolutionized the field of by enabling the generation of realistic and visually pleasing high-resolution images, and here SRGAN is introduced. There is two key components of SRGAN which make the model work better at containing the texture of images, the loss function and architecture. The architecture of the model is designed as following figure

The upper part is the generator component, which will produce the high quality reconstructed images. The core of the network of generator is the residual block layout, which includes two convolutional layers with small 33 kernels and 64 feature maps. The residual connection enable the model to keep the information from different dimension, known as the texture. The resolution of input images are also increased by the sub-pixel convolutional layer by Shi et al.[41] The discriminator is defined following the guide summarized by Radford et al.[51] The resulting 512 feature maps are followed by two dense layers and a final sigmoid activation function to obtain a probability for sample classification.

The most common pixel-wise loss function such as mean square error function are difficult to recover the texture as they lead to pixel-wise optimal solution. Typically, the images are over-smooth thus have lower perceptual quality[52, 53, 54]. According to Johnson et al. and Bruna et al.[55, 53] a new loss function is defined based on perceptual similarity:

$$l^{SR} = l_X^{SR} + 10^{-3} l_{GEN}^{SR}$$

The perceptual loss is formulated as the weighted sum of content loss(l_X^{SR}) and adversarial loss($10^{-3} l_{GEN}^{SR}$). The content loss is defined according to the comparison between the result after convolutional layers and the original HR images. The adversarial loss is defined based on the probabilities of that discriminator regards the re-constructed images as natural images.

E. Transformer

Transformer has recently gained increasing exposure and usage in image processing field. For up-scaling images, a low-resolution image is required to feed into the transformer model, which learns to generate a corresponding high-resolution image. The transformer contains two sub-layers, namly

multi-head self-attention layer and feed-forward layer. The self-attention mechanism operates on an input sequence by computing the dot product of the query and key vectors for each element in the sequence. This yields a matrix of dot product values. The dot product values are then divided by (dk) and passed through a softmax function, resulting in a weight distribution for each query. These weights represent the importance or relevance of each element in the sequence with respect to the corresponding query.

The weighted values are multiplied with the value matrix V, producing the final output of the self-attention mechanism.The equation goes like this,

$$\text{Att}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \qquad (12)$$

After the self-attention, there comes the feed-forward layer, a transformer decoder responsible for generating the up-scaled image. It takes the output from the encoder as input and generates a sequence of higher-resolution patches. The decoder also includes masked self-attention to ensure that the model considers only the previously generated patches, preventing information leakage from future patches.[56] Those higher-resolution patches are stitched together to form the final up-scaled image after the decoder.

During training, the model is typically provided with pairs of low-resolution and high-resolution images. It learns to minimize the difference between the generated high-resolution image and the ground truth high-resolution image through a loss function, such as mean squared error or perceptual loss. The model's parameters are optimized through back-propagation[57], updating the weights to improve its ability to reconstruct high-resolution details.

Once training is complete, the transformer-based image up-scaling model can take a new low-resolution image as input and generate a high-resolution version with enhanced details, such as sharper edges and more intricate textures.

The transformer decomposes the video sequence into two dimensions, time and space, and then converts this data into self-attention calculations to reduce complexity. The self-attention mechanism allows the model to focus on different parts of the input image when processing each position. It computes the attention weights for each position in the input by comparing it to all other positions. These attention weights indicate the relevance of each position with respect to the others. [58]

Transformer enables the model to capture relationships between different elements of the sequence have a significantly lower number of parameters, regardless of their positions. For CNNs, The number of parameters in a convolutional layer depends on the size of the filters and the number of channels in the input and output. As a result, CNNs can have a large number of parameters, especially in deeper architectures. On the other hand, by using self-attention layer which would enable the transformer to gather information from distant parts of the input sequence without using convolutions or parameter sharing, transformer uses projections to transform the input embeddings into key, query, and value vectors.[59] Those projections, which usually linear projections, have a significantly lower number of parameters. Therefore, transformer has less parameter, less computational complexities with better performance compared with traditional CNN.

Overall, by leveraging the power of self-attention mechanisms, transformers can effectively capture long-range dependencies and learn complex patterns in the image data. This allows them to generate visually appealing up-scaled images with improved resolution and enhanced details compared to traditional interpolation-based methods. However, transformer may struggle with capturing spatial information and local patterns that are crucial in certain computer vision tasks. They also require large amounts of data to generalize effectively. Due to their high parameter efficiency, transformers may not perform as well as CNNs on tasks with limited training data too.

VII. Implemented Algorithm Result and Evaluation

A. Testing Results on Images

Algorithms discussed in this literature review have been implemented in Python or MATLAB, where possible, to better compare and evaluate them. The performance of implemented algorithms, when upscaling with a factor of 4, have

been tested in a number of images from dataset DIV2K [60]. It is decided to performance the testing on images first, because the algorithms implemented are all single-frame based and their performance on super-resolving images are indicative of their performance on videos.

The input to the algorithms are low resolution images obtained using Matlab imresize function with default settings and a downscale factor of 4. The output results are compared with original high resolution images to calculate peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM). The execution time is also measured to give an indication of the computational cost.

Fig. 17. The original High Resolution Image 0882 from DIV2K

Figure 17 is the image 0882 from dataset DIV2K [60] and one of the image tested. The detailed cropped images showing the performance of each algorithm.

(a) Original (b) Bicubic Interpolation (c) Lanczos Interpolation

Fig. 18. Bicubic and Lanczos Interpolation

The results recorded in table II and shown in figure 19 for all wavelet-based methods are only the results obtained using the most basic wavelet, which is the haar wavelet for DWT

(a) Original HR (b) Wavelet Zero Padding (c) DWT with SWT

(d) DWT with NEDI (e) DTCWT with Lanczos

Fig. 19. Wavelet Based Methods with Haar and Biorthogonal 1.3 Wavelet

(a) Original HR (b) Wavelet Zero Padding (c) DWT with SWT

(d) DWT with NEDI (e) DTCWT with Lanczos

Fig. 20. Wavelet Based Methods with CDF 9/7 and Antonini 9/7 wavelet

and biorthogonal 1.3 wavelet for DT-CWT. Since wavelet transform with other wavelets (at least with the PyWavelets library used) produce images with slightly different dimensions compared to the original one, making it impossible to make meaningful comparison with the original without extra resizing. To better evaluate the performance of wavelet based algorithms, where possible, super-resolved images have also been produced using the same methods but with CDF 9/7 wavelet for DWT, also known as biorthogonal 4, 4 wavelet, which

allows for better image reconstruction than other conventional wavelets [61]; and with Antonini 9/7 wavelet for DT-CWT. The execution time for implementation with CDF 9/7 and Antonini 9/7 wavelet is approximately the same and omitted for presentation in this review.

As a result of the simple wavelet used, the output images shown in 19 have clear pixelated texture and slightly lower PSNR and SSIM even when compared with bicubic and lanczos interpolation, as shown in table II. This is particularly the case for the wavelet zero padding method, as shown in figure 19. The DWT with SWT method also produces unsatisfactory upscaled image, which have blurred edges and textures. Nonetheless, for DWT with NEDI and DTCWT with Lanczos, it is clear from the output images shown in 19 and 20 that they produce more distinct and sharper edges compared to bicubic and lanczos interpolation. The textures, except the pixelated appearance due to usage of very basic wavelets, are maintained well and realistic, unlike the NEDI that over-smooths the edges and results in drawing-like textures. These methods also do not produce zig-zag or fuzzy artefacts like the sobel based method and the sobel based method with zig-zag suppression. Moreover, the image super-resolved by wavelet zero padding with CDF 9/7 wavelet have much less pixelated texture and better visual quality compared to wavelet zero padding result with haar wavelet, which illustrates the impact of exact wavelet used on the algorithm performance.

The execution time in software for wavelet-based methods is longer than bicubic and lanczos interpolation, but comparable to all other adaptive interpolation methods. In the process of executing these algorithms, significantly much more time is taken for the interpolation of sub-bands, rather than the wavelet transforms, partcularly for the case of DWT with NEDI. Nonetheless, when implemented in FPGA, the wavelet transform may takes a much longer time. Further investigation should be made to evaluate wavelet-based methods, when implementing with different wavelets and in FPGA.

The NEDI method shown in figure 21, by virtue of its edge-directed approach, prioritizes the preservation of high-frequency details such as edges and textures during the upscaling process.

(a) Original HR (b) NEDI (c) NEDI with Canny and Bicubic

(d) Local Structure (e) Sobel Based (f) Sobel Based with Zig-Zag Suppression

Fig. 21. Adaptation Based on Local Information Interpolation

When applied to video upscaling, it tends to provide a visually pleasing output with sharp edges and detailed textures. By changing the window size in the NEDI code, we can get slightly different results. Smaller the window size, stronger the contrast at the edges; larger the window size, smoother the edges. However NEDI algorithm requires quite intense computation due to the inverse matrix computation applied on each pixel. To solve this problem NEDI with Canny and Bicubic is implemented. NEDI algorithm is only applied onto pixels detected as edges by Canny edge detection, while the rest of the input image is resized with bicubic interpolation. In this way a balance of edge quality and computation speed is achieved.

The Sobel based method takes a relatively little amount of time to execute in software, while the local structure execution takes way too long. As shown in figure 21, the Sobel based method does preserve the edges relatively well, yet it produces a lot of zig-zag artefacts along the edges, resulting in relatively low PSNR and SSIM. Using zig-zag suppression method can indeed reduce zig-zag effects and make the output image smoother, however, the output images can be over-smoothed, resulting in blurry and unrealistic textures, also causing lower PSNR and SSIM. Changing and picking the suitable canny edge threshold in this combined process could ideally improve the re-

sults, yet a method for determining the optimum threshold should be developed.

(a) Original HR (b) ESRGAN

Fig. 22. ESRGAN performance

As the Figure 22 shows, the quality of image super-resolved by ESRGAN is superior compared to other algorithms implemented, with accurate and sharp edge as reflected from the highest PSNR and visual quality. However, similar to all other algorithms' results, the texture of the pollen at the centre of the image is still missing. Besides, the model takes about 10 seconds to process the image; it also takes an enormous amount of time to train the model. It needs further optimisation and utilization of parallel computing for acceleration, particularly when implemented on FPGA.

Method	PSNR(dB)	SSIM(%)	Software Execution Time(s)
Bicubic	31.57	92.84	**0.303**
Lanczos	31.69	92.97	16.876
Wavelet Zero Padding	28.16	86.67	0.523
DWT with SWT	21.63	60.61	1.264
DWT with NEDI	30.76	91.07	740.177
DT-CWT with Lanczos	30.39	90.38	49.279
NEDI	30.26	94.78	136.104
NEDI with Canny edge and Bicubic Interpolation	31.04	**95.49**	3.331
Local Structure	26.16	88.89	1052.629
Sobel Based Interpolation	28.84	90.69	53.561
Sobel Based Interpolation with Zig-Zag Suppression	27.16	89.17	65.04
ESRGAN	**32.95**	91.10	9.437

TABLE II

RESULTS OF IMPLEMENTED ALGORITHMS

B. Testing Results on Videos

The procedure of testing and evaluating algorithms' performance on videos is similar to the testing on images; it is essentially applying the same code to every frame captured from the videos. We choose a video of 2 seconds which includes 60 frames in total. The low-resolution video, which is down-sampled by 4 from the original using Pillow's bicubic resize, will be input into each of the chosen algorithms, the 4x upscaled output will be compared with the original high-resolution video frame by frame. The average PSNR, SSIM and the execution time per frame over the video will be calculated, using the same code as the testing on images.

Fig. 23. One Random Frame of Original High Resolution London Video (1920x1080)

Fig. 24. One Random Frame of Original High Resolution Las Vegas Video (3840x2160)

Method	Average PSNR(dB)	Average SSIM(%)	Software Execution Time(s) Per Frame
Bicubic	26.17	63.98	**0.0018**
DT-CWT with Lanczos	25.84	60.70	30.57
NEDI with Canny edge and Bicubic Interpolation	25.83	62.31	8.868
Local Structure	23.36	52.75	236.68
ESRGAN	**27.8539**	**83.03**	0.997

TABLE III

EVALUATION BASED ON VIDEO TESTING (LONDON VIDEO) (1920X1080)

As shown from table III, when upscaling video instead of images, all the algorithms' performance reduce significantly. When comparing with the testing results for image 0882 from DIV2K data set, the PSNR is lowered by 4.61 dB on average for all the chosen algorithms. The least decrease is 2.80 dB for local structure based method, and the greatest decrease is 5.40 dB for bicubic interpolation. The SSIM is lowered by 27.19% on average

Method	Average PSNR(dB)	Average SSIM(%)	Software Execution Time(s) Per Frame
Bicubic	27.26	78.61	**0.0050**
DT-CWT with Lanczos	26.83	76.07	121.60
NEDI with Canny edge and Bicubic Interpolation	26.75	77.04	19.75
Local Structure	-	-	-
ESRGAN	**28.2529**	**86.46**	4.205

TABLE IV
EVALUATION BASED ON VIDEO TESTING (LAS VEGAS VIDEO) (3840X2160)

for all the chosen algorithms. The least decrease is 8.07% for ESRGAN, and the greatest decrease is 36.14% for local structure based method. This decrease in PSNR and SSIM, on one hand, is caused by the inevitable existence of blurred frames in videos due to motion, which makes it difficult for edge adaptive algorithms to explicitly or implicitly detect and correctly preserve edges. On the other hand, the original HR video used only has 1080p resolution, which makes the 4x downscaled LR video unable to preserve enough edges and textures, for upscaling algorithms to work with. This is proven in the testing with the second video shown in figure 24, which has a higher resolution and consequently all the algorithms have considerably better performance in terms of PSNR, SSIM and visual quality, with an inevitably longer execution time. (The testing with local structure is omitted due to the overly long processing time with the higher resolution video.)

The execution time per frame for video is considerably lowered compared to the execution time for single image. The execution time is lowered by 167.6 seconds on average for all the chosen algorithms. This reduction could be caused by the usage of different platform, since the testing on video is done on local desktop through the application software whereas the testing on images is done on Google Colab; this reduction could also be caused by the fact that memory allocation is needed less frequently when processing videos compared to images, due to the repeated usage of same functions and variables, as well as due to the temporal coherence of video frames that causes repetition of data representing the frames. For improved NEDI, the execution time increases by 5.53 seconds. This is caused because in image testing, the improved NEDI is tested directly in MATLAB; in video testing, it involves extra stages of reading and writing images and communicating between Python application and MATLAB program.

Overall, it is ESRGAN performing the best on upscaling videos and the execution time will be much shorter than working on single image because that the model will learn the feature of the frames, so that the similar frames can be calculated faster. However, ESRGAN has the higher computational cost (ALM usage) of all algorithms, when implemented in FPGA.

VIII. CONCLUSIONS

MISR approaches, including multi-frame image registration and motion estimation, are very powerful and can be very effective, however, almost unanimously slow due to the difficulty involved in mapping images onto one another. The state of the art in this field is learning based optical flow estimation with methods such as RAFT achieving accuracy to within several pixels []. This area is a highly active area of research and perhaps in the future, such methods will be worth exploring for commercially viable video up-scaling. However, as it stands, we will be focusing on single-image super-resolution.

Classical interpolation techniques such as bilinear and bicubic interpolation have served as the foundation approach of up-scaling images for many years. These methods work by calculating the values of new pixels based on neighboring pixels in the original image. While they are relatively simple and computationally efficient, they may produce results that lack fine details and exhibit blurring or aliasing artifacts.

Fourier domain methods provide some excellent insights into the field of super-resolution and the fundamental theory behind SR is based on them. However, these methods are simply not effective nor efficient enough to be commercially viable. Wavelet transform-based methods offer a much more promising prospect. Wavelet zero padding allows for fast super-resolution, with execution time in software comparable to bicubic and Lanczos interpolation, while having only slightly worse

performance. Other more complicated DWT or DT-CWT based algorithms indeed have more execution time and computational complexity compared to bicubic and lanczos interpolation, yet yield significantly sharper edges without apparent artefacts and noise. The DT-CWT with bicubic/lanczos interpolation method has a particularly low executing time in software as shown in table II. Nonetheless, this may not reflect the latency or computational complexity when implemented in FPGA, and only implementation with the basic haar wavelet has been made in this review, which limits the quality of super-resolved video frames, further investigation should be made into these areas.

On the other hand, some more novel edge interpolation methods such as the local structure method discussed and Sobel based method discussed are similar in the sense that they calculate the coefficients of interpolation adaptively. Both of these approaches theoretically retain the speed of the classical interpolation methods but they are significantly more effective, with higher PSNR and SSIM. They can produce better and clearer images than traditional interpolation, thus should be further researched and developed.

As for the neural network, it does show great performance in both keep sharp edge and fine texture compared to other algorithms with the highest PSNR, which can be seen as a feasible methods. However, there are still issues existing with it, such as the much higher cost. A more serious issue will be the problem of unstable background of the videos in the actual application. Because the feature of GAN model, it will generate new images every frame, which is difficult for model to keep the background consistent.

References

[1] Hong Chang, Dit-Yan Yeung, and Yimin Xiong. "Super-resolution through neighbor embedding". In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* IEEE. 2004, pp. I–I.

[2] Shen-Chuan Tai, Jiun-Jie Huang, and Peng-Yu Chen. "A Super-Resolution Algorithm Using Linear Regression Based on Image Self-Similarity". In: *2016 International Symposium on Computer, Consumer and Control (IS3C)*. IEEE. 2016, pp. 275–278.

[3] Robert Keys. "Cubic convolution interpolation for digital image processing". In: *IEEE transactions on acoustics, speech, and signal processing* 29.6 (1981), pp. 1153–1160.

[4] Xun Wang. "Interpolation and Sharpening for Image Upsampling". In: *2022 2nd International Conference on Computer Graphics, Image and Virtualization (ICCGIV)*. 2022, pp. 73–77. DOI: 10.1109/ICCGIV57403.2022.00020.

[5] Jagyanseni Panda and Sukadev Meher. "A New Residual Image Sharpening Scheme for Image Up-Sampling". In: *2022 8th International Conference on Signal Processing and Communication (ICSC)*. 2022, pp. 244–249. DOI: 10.1109/ICSC56524.2022.10009436.

[6] J. L. Harris. "Diffraction and Resolving Power*". In: *J. Opt. Soc. Am.* 54.7 (July 1964), pp. 931–936.

[7] Ernst Adolph Guillemin. *The mathematics of circuit analysis*. John Wiley New York, 1949.

[8] RW Gerchberg. "Super-resolution through error energy reduction". In: *Optica Acta: International Journal of Optics* 21.9 (1974), pp. 709–720.

[9] Roger Y Tsai and Thomas S Huang. "Multiframe image restoration and registration". In: *Multiframe image restoration and registration* 1 (1984), pp. 317–339.

[10] Forrest Hoffman. "An introduction to Fourier theory". In: *Extraído el* 2 (1997).

[11] Hasan Demirel and Gholamreza Anbarjafari. "IMAGE Resolution Enhancement by

Using Discrete and Stationary Wavelet Decomposition". In: *IEEE Transactions on Image Processing* 20.5 (2011), pp. 1458–1460. DOI: 10.1109/TIP.2010.2087767.

[12] Zhi-Song Liu, Wan-Chi Siu, and Jun-Jie Huang. "Image super-resolution via hybrid NEDI and wavelet-based scheme". In: *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. 2015, pp. 1131–1136. DOI: 10.1109/APSIPA.2015.7415447.

[13] Yasutaka Matsuo and Shinichi Sakaida. "Super-resolution for 2K/8K television using wavelet-based image registration". In: *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. 2017, pp. 378–382. DOI: 10.1109/GlobalSIP.2017.8308668.

[14] Chang-Ming Lee et al. "Super-resolution reconstruction of video sequences based on wavelet-domain spatial and temporal processing". In: *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. 2012, pp. 194–197.

[15] H. Zheng, A. Bouzerdoum, and S. L. Phung. "Wavelet based nonlocal-means super-resolution for video sequences". In: *2010 IEEE International Conference on Image Processing*. 2010, pp. 2817–2820. DOI: 10.1109/ICIP.2010.5651488.

[16] Chia-Chun Hsu, Jian-Jiun Ding, and Yih-Cherng Lee. "Efficient edge-oriented based image interpolation algorithm for non-integer scaling factor". In: *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. 2017, pp. 1156–1159. DOI: 10.1109/APSIPA.2017.8282202.

[17] Hasan Demirel and Gholamreza Anbarjafari. "Satellite Image Resolution Enhancement Using Complex Wavelet Transform". In: *IEEE Geoscience and Remote Sensing Letters* 7.1 (2010), pp. 123–126. DOI: 10.1109/LGRS.2009.2028440.

[18] Sara Izadpanahi and Hasan Demirel. "Multiframe super resolution using edge directed interpolation and complex wavelet transform". In: *IET Conference on Image Processing (IPR 2012)*. 2012, pp. 1–5. DOI: 10.1049/cp.2012.0447.

[19] Pilla Jagadeesh and Jayanthi Pragatheeswaran. "Image resolution enhancement based on edge directed interpolation using dual tree fffdfffdfffd Complex wavelet transform". In: *2011 International Conference on Recent Trends in Information Technology (ICRTIT)*. 2011, pp. 759–763. DOI: 10.1109/ICRTIT.2011.5972260.

[20] M. Irani and S. Peleg. "Super resolution from image sequences". In: *[1990] Proceedings. 10th International Conference on Pattern Recognition*. Vol. ii. 1990, 115–120 vol.2.

[21] Bruce D Lucas and Takeo Kanade. "An iterative image registration technique with an application to stereo vision". In: *IJCAI'81: 7th international joint conference on Artificial intelligence*. Vol. 2. 1981, pp. 674–679.

[22] WenYi Zhao and Harpreet S Sawhney. "Is super-resolution with optical flow feasible?" In: *Computer Vision ECCV 2002: 7th European Conference on Computer Vision Copenhagen, Denmark, May 28–31, 2002 Proceedings, Part I 7*. Springer. 2002, pp. 599–613.

[23] Michal Irani and Shmuel Peleg. "Motion analysis for image enhancement: Resolution, occlusion, and transparency". In: *Journal of visual communication and image representation* 4.4 (1993), pp. 324–335.

[24] Heng Su, Ying Wu, and Jie Zhou. "Super-Resolution Without Dense Flow". In: *IEEE Transactions on Image Processing* 21.4 (2012), pp. 1782–1795.

[25] David G Lowe. "Distinctive image features from scale-invariant keypoints". In: *International journal of computer vision* 60 (2004), pp. 91–110.

[26] Jun-Yong Kim, Rae-Hong Park, and Seungjoon Yang. "Super-resolution using POCS-based reconstruction with artifact reduction constraints". In: *Visual Communications and Image Processing 2005*. Vol. 5960. SPIE. 2005, pp. 1810–1818.

[27] John Canny. “A computational approach to edge detection”. In: *IEEE Transactions on pattern analysis and machine intelligence* 6 (1986), pp. 679–698.

[28] S. Carrato, G. Ramponi, and S. Marsi. “A simple edge-sensitive image interpolation filter”. In: *Proceedings of 3rd IEEE International Conference on Image Processing*. Vol. 3. 1996, 711–714 vol.3.

[29] Xin Li and M.T. Orchard. “New edge-directed interpolation”. In: *IEEE Transactions on Image Processing* 10.10 (2001), pp. 1521–1527.

[30] Xin Li and Michael T Orchard. “Edge-directed prediction for lossless compression of natural images”. In: *IEEE Transactions on image processing* 10.6 (2001), pp. 813–817.

[31] Hang Sun et al. “A Novel Hardware-Based UHD Video Up-Scaler Based on Local Structure Estimation”. In: Springer.

[32] Wanli Chen, Ya Jun Yu, and Hongjian Shi. “An Improvement of Edge-Adaptive Image Scaling Algorithm Based on Sobel Operator”. In: *2017 4th International Conference on Information Science and Control Engineering (ICISCE)*. 2017, pp. 183–186. DOI: 10.1109/ICISCE.2017.48.

[33] Wanli Chen and Hongjian Shi. “An Edge Based Adaptive Interpolation Algorithm for Image Scaling”. In: 2018.

[34] Wikipedia contributors. *Comparison gallery of image scaling algorithms — Wikipedia, The Free Encyclopedia*. [Online; accessed 28-June-2023]. 2023. URL: https://en.wikipedia.org/w/index.php?title=Comparison_gallery_of_image_scaling_algorithms&oldid=1137647484.

[35] K Chaitanya Pavan Tanay et al. “Fast video super resolution using deep convolutional networks”. In: (2017), pp. 1–6. DOI: 10.1109/ICIIECS.2017.8276067.

[36] Adam Paszke et al. “Automatic differentiation in pytorch”. In: (2017).

[37] Martín Abadi et al. “Tensorflow: Large-scale machine learning on heterogeneous distributed systems”. In: *arXiv preprint arXiv:1603.04467* (2016).

[38] Yongwoo Kim, Jae-Seok Choi, and Munchurl Kim. “A Real-Time Convolutional Neural Network for Super-Resolution on FPGA With Applications to 4K UHD 60 fps Video Services”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 29.8 (2019), pp. 2521–2534. DOI: 10.1109/TCSVT.2018.2864321.

[39] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. “Accurate Image Super-Resolution Using Very Deep Convolutional Networks”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 1646–1654. DOI: 10.1109/CVPR.2016.182.

[40] Chao Dong et al. “Image Super-Resolution Using Deep Convolutional Networks”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.2 (2016), pp. 295–307. DOI: 10.1109/TPAMI.2015.2439281.

[41] Wenzhe Shi et al. “Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 1874–1883. DOI: 10.1109/CVPR.2016.207.

[42] Wikipedia contributors. *YCbCr — Wikipedia, The Free Encyclopedia*. [Online; accessed 23-May-2023]. 2023. URL: https://en.wikipedia.org/w/index.php?title=YCbCr&oldid=1147788722.

[43] Chao Dong, Chen Change Loy, and Xiaoou Tang. “Accelerating the super-resolution convolutional neural network”. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*. Springer. 2016, pp. 391–407.

[44] Andrew G Howard et al. “Mobilenets: Efficient convolutional neural networks for mobile vision applications”. In: *arXiv preprint arXiv:1704.04861* (2017).

[45] Ian J. Goodfellow et al. *Generative Adversarial Networks*. 2014. arXiv: 1406.2661 [stat.ML].

[46] Han Zhang et al. "StackGAN: Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 5908–5916. DOI: 10.1109/ICCV.2017.629.

[47] Jingwen Chen et al. "Image Blind Denoising with Generative Adversarial Network Based Noise Modeling". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 3155–3164. DOI: 10.1109/CVPR.2018.00333.

[48] Qingsong Yang et al. "Low-Dose CT Image Denoising Using a Generative Adversarial Network With Wasserstein Distance and Perceptual Loss". In: *IEEE Transactions on Medical Imaging* 37.6 (2018), pp. 1348–1357. DOI: 10.1109/TMI.2018.2827462.

[49] Yunjey Choi et al. "StarGAN: Unified Generative Adversarial Networks for Multi-domain Image-to-Image Translation". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8789–8797. DOI: 10.1109/CVPR.2018.00916.

[50] Christian Ledig et al. *Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network*. 2017. arXiv: 1609.04802 [cs.CV].

[51] Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks". In: *arXiv preprint arXiv:1511.06434* (2015).

[52] Michael Mathieu, Camille Couprie, and Yann LeCun. "Deep multi-scale video prediction beyond mean square error". In: *arXiv preprint arXiv:1511.05440* (2015).

[53] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. "Perceptual Losses for Real-Time Style Transfer and Super-Resolution". In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe et al. Cham: Springer International Publishing, 2016, pp. 694–711. ISBN: 978-3-319-46475-6.

[54] Alexey Dosovitskiy and Thomas Brox. "Generating images with perceptual similarity metrics based on deep networks". In: *Advances in neural information processing systems* 29 (2016).

[55] Joan Bruna, Pablo Sprechmann, and Yann LeCun. "Super-resolution with deep convolutional sufficient statistics". In: *arXiv preprint arXiv:1511.05666* (2015).

[56] Haiyong Wang and Kai Jiang. "Research on Image Super-Resolution Reconstruction Based on Transformer". In: *2021 IEEE International Conference on Artificial Intelligence and Industrial Design (AIID)*. 2021, pp. 226–230. DOI: 10.1109/AIID51893.2021.9456580.

[57] Chongjun Ye et al. "A Super-resolution Method of Remote Sensing Image Using Transformers". In: *2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*. Vol. 2. 2021, pp. 905–910. DOI: 10.1109/IDAACS53288.2021.9660904.

[58] Minyan Zheng, Jianping Luo, and Wenming Cao. "Video Super-Resolution Based on Spatial-Temporal Transformer". In: *2021 IEEE 7th International Conference on Cloud Computing and Intelligent Systems (CCIS)*. 2021, pp. 403–407. DOI: 10.1109/CCIS53392.2021.9754604.

[59] Kai Chen, Chao Liu, and Yongsheng Ou. "Channel Attention based Network for LiDAR Super-resolution". In: *2021 China Automation Congress (CAC)*. 2021, pp. 5458–5463. DOI: 10.1109/CAC53003.2021.9727846.

[60] Eirikur Agustsson and Radu Timofte. "NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. July 2017.

[61] K Kishore Kumar et al. "Resolution enhancement using DWT and SWT by Fusion techniques with watermarking". In: *2014*

IEEE International Conference on Computational Intelligence and Computing Research. 2014, pp. 1–5. DOI: 10.1109/ICCIC.2014.7238554.