Egyptian Informatics Journal 23 (2022) 159–172

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

An improved Image Interpolation technique using OLA e-spline

Jagyanseni Panda [a,*], Sukadev Meher [b]

[a] Dept. of Electronics and Communication Engineering, National Institute of Technology Rourkela, (Study leave from SOA University), Rourkela 769008, Odisha, India
[b] Dept. of Electronics and Communication Engineering, National Institute of Technology Rourkela, Rourkela 769008, Odisha, India

ARTICLE INFO

Article history:
Received 16 April 2021
Revised 29 August 2021
Accepted 22 October 2021
Available online 19 November 2021

Keywords:
Artifacts
Blurred image
Edge expansion
Image upscaling
Unsharp masking

ABSTRACT

Image upscaling aims to increase the resolution and size of a low resolution (LR) image in order to generate a high resolution (HR) image of high frequency (HF). There are several polynomial methods for obtaining a sharpened, upscaled HR image. The interpolated pixel is measured using a weighted average of the neighboring pixels within the image grid that blur at HF regions in these methods. Edge degradation is also caused by other edge-directed and learning-based upscaling methods, which produce blurring artifacts. A novel approach is proposed to fill these gaps. Using the concept of unsharp masking (USM), the LR image is blurred adaptively based on the region's local variance. The sharpened high pass filtered (HPF) image is then obtained by subtracting the adaptively blurred image from the LR image. According to USM, the HPF image is combined with the LR image via a gain factor optimized using the cuckoo search (CS) algorithm. To compensate for the loss caused by upscaling, this pre-processing step is performed prior to interpolation. Aside from that, the edge of the B-spline interpolated image is detected and expanded. Edge expansion of the upscaled image is performed to further restore the HF details and reduce zigzag artifacts introduced by upscaling while also preserving the edge boundary. The proposed method outperforms the Lanczos, Bicubic, and Bilinear schemes in terms of peak signal to noise ratio (PSNR) gain of 3.475, 8.3839, and 8.075 dB, respectively. In terms of performance, this method outperforms state-of-the-art techniques both objectively (PSNR and SSIM) and subjectively (visual quality).

1. Introduction

Images and videos play an important role in multimedia communication, which is an important part of the digital world. However, the scarcity of bandwidth poses a problem in the transmission of images and videos via multimedia communication. Narrow bandwidth is ineffective for transferring large amounts of data in image and video communication. To address this issue, the current practice is to scale down the image or video size at the encoder's end and upscale the received, downscaled image at the decoder's end. However, a significant loss of image quality occurs during this process. A low-quality camera lens is another factor that affects image quality. The current paper proposes a method for retrieving the original image's details and restoring them in the upscaled image. As a result, the final result is a high resolution (HR) image with discrete boundaries and rich texture details, as well as high frequency (HF) region details. HR images have sharpened edges and a richness in texture details. To be more specific, both the object and the background are distinguishable in an HR image, with no blurs. It also lacks ringing artifacts in the image's HF regions. HR images are currently finding widespread use in medical imaging, satellite imaging, security services (face recognition), and video surveillance systems such as drones are all possibilities. Image upscaling is a difficult process. To begin, the object's and background's distinct boundaries must be preserved while the image's edge sharpness and texture richness are enhanced. Second, in order to perform image applications in real-time, the upscaling method should be fast and easy. Image upscaling is also known as image interpolation, zooming, or enlargement.

This paper proposes a new image interpolation technique for producing an HR image. The following are the major contributions of the proposed work:

* Corresponding author.
E-mail addresses: jagyansenipanda12@gmail.com (J. Panda), smeher@nitrkl.ac.in (S. Meher).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.eij.2021.10.002

- The smooth areas and the image's edges or HF regions are treated differently in this pre-processing step. It is also used prior to up-sampling to eliminate the artifacts introduced by interpolation.
- A Local adaptive (LA) filter is used to blur the image adaptively based on the local variance. The filtered image retains the outline and smooth-area texture of the original LR image. It is used to highlight the HF of an image.
- The difference between the LR and adaptively blurred images is referred to as the high pass filtered (HPF) image.
- Following filtration, the LR image and HPF images are combined using an optimum scale factor by using the Cuckoo search (CS) optimization algorithm to obtain a restored LR image, according to the unsharp masking (USM) concept.
- The restored LR image is then upscaled using edge-preserving spline (e-spline), retaining all of the previously retrieved details. The e-spline preserves the shape of the edge while avoiding zigzag artifacts. Our method is simple and adaptable, relying on simple filtering techniques for different image scales.

To produce a final sharpened HR image, this method restores some details of the LR image that are supposed to be lost after interpolation. The analysis of the results shows that the proposed method consistently generates a high-quality image while preserving the information of both edges and texture. Our method is very simple and adaptable, relying on simple filtering techniques for different image scales. This is a pre-processing technique that improves the edges and fine details in the LR image, which would lose these HF components significantly if up-sampled, resulting in high quality in the upscaled HR version of the image. To overcome the blur and zigzag artifacts caused by upscaling, the LR image is further enlarged using B-spline interpolation followed by edge expansion. As a result, the proposed method is referred to as optimized local adaptive edge-preserving spline (OLA e-spline). The remainder of the paper is structured as follows. The application, as well as the specialty of the proposed algorithm, are discussed in the first section. Section 2 contains a review of the literature. Section 3 describes the proposed algorithm, while Section 4 presents experimental results. Finally, conclusion is presented in Section 5.

2. Literature review

In real-time applications to increase the resolution of LR images, various conventional interpolation methods like Bilinear [1], Bicubic [2], Spline [3], and Lanczos [4] are proposed. These methods are simple because the value of the interpolated pixel is calculated by averaging or taking a convolution sum of the neighboring pixels. While performing image interpolation using these polynomial-based methods, one often gets blurred edges or staircase artifacts in the resultant image. To overcome these problems, improved edge interpolation methods are proposed in [5–8]. New edge directed interpolation (NEDI) [5] is one local adaptive edge directed interpolation. In this method, the covariance of the HR image can be computed by using duality over LR image covariance. And the missing-edge pixels are interpolated through the edge-directed method and non-edge pixels are interpolated with the bilinear method. The method proposed in [6] is an upgraded version of cubic convolution (CC), i.e. it is Directional CC (DCC). Depending on the direction of the missing pixel, the missing pixel will be interpolated in that direction using CC. One new form of the edge-directional method based on linear minimum mean square estimation (LMMSE), is presented in [7]. The missing pixel is interpolated in two orthogonal directions using the pixel value of its neighbors in these directions. The final interpolated result is obtained by fusing the two interpolated values by using LMMSE. Because this method performs edge interpolation along the edge rather than across the edge, boundary information is lost. To determine missing pixels in [8] different gaussian kernels with different standard deviation (SD) are applied. Its SD is determined using the interpolation window SD. This interpolated image's subjective quality is higher than its objective quality. New approaches of edge restoration are introduced in [9,10] where edge error is used to restore the blur artifacts after interpolation. However, when compared to other conventional methods, the flexibility of these methods for different scaling factors is less. To further improve the quality of interpolated results, an alternative method is proposed by Giachetti et al. [11] i.e. the Iterative Curvature Based Interpolation (ICBI). The missing pixels are interpolated in edge-directions first. Then, the interpolated values are refined by the smoothening of the second-order directional changes keeping the pixel value constant. However, these methods gave rise to a different issue altogether: the interpolation algorithms cause highly smoothened images; thus resulting in the loss of details. Because of the two-pass approach, the computational time in ICBI is longer. So some methods focus on the computation time of the interpolation algorithm. One such fast interpolation method is proposed in [12], where the missing pixels are interpolated by using weighted averaging of known pixels in the image grid. In this, the distance between known and unknown pixels is used as a weight for interpolation. The error occurred as a result of the inaccuracy of the weight prediction. To make interpolation algorithms suitable for real-time applications, the regression-based upscaling methods are approached in [13,14]. A changed version of the NEDI, like autoregressive (AR) is proposed in [13], where a moving least-square method is used. It is also a local interpolation technique, where each LR patch is searched for the nearest neighborhood patches and maps the LR patches to another space for generating HR image. Zhang et al. [14] have proposed an improved AR algorithm to get better results, both subjectively and objectively.

To reduce the complexity of the above regression methods, an error-based interpolation method is proposed in [15], which uses the error in between the two sampling points in edge-direction, to interpolate the missing pixels. However, all this method cannot generate sharpened up-scaled images. To overcome this issue, the gradient details of an LR image are used in [16,17] while upscaling it. In [16], the missing pixels are interpolated by using the gradient features of the local, neighboring pixels, for proper reconstruction of the HR image. Similarly, the interpolation method proposed in [17] uses the gradient sharpening transformation method to get an HR image. However, these gradient-based upscaling methods produce false edges in the resultant images. So, to overcome this problem, Zhu et al. have proposed [18] a learning-based method. It utilizes the geometric structural similarity between the LR patch and the HR patch, alongside using a directional gradient as a regularizing factor, to improve the quality of restoration. Zhang et al. have used multiple-linear mapping in [19] that directly maps an LR patch to multiple HR patches. In, [20] variable no. of training patches are used for HR image reconstruction instead of fixed patches. Overall, two drawbacks are noted in these learning-based methods. First, it has a longer running time. Second, its image enlargement works for particular-scale factors only. If the scaling factor changes, the learning method generates artifacts in the texture region and the dictionary-learning methods are very complex. There are some reconstruction-based methods, as discussed in [21,22]. Reconstruction involves the utilization of gradient information from different images a prior, to produce a sharper HR image [21]. Similarly, in [22] the total variation and gradient of the image act as a before reconstruct a sharper image. But as the scaling factor increases, the blurring occurs in the texture part of the image. For this, an HR image looks unnatural. To solve the

issue, Khan et al. [23] have used adjacent pixels slope to generate HR image with texture details. For a complex-textured image, a new reconstruction-based interpolation method is proposed in [24]. But the running time of this method is very high. One new, graph-based interpolation method is proposed in [25] to preserve more detailed information in the HR image. In this method, information is propagated from a known pixel to an unknown pixel, alongside preserving the information of the sharp edges and continuous texturing. All the above methods fall short of producing sharpened HR images, as they cannot restore the HF degradation happening during interpolation. So, a new method is proposed here, which is based on preserving the edge and texture, to achieve better image quality in interpolation. It also preserves natural looks in the HR image.

3. Problem formulation

The primary goal of image up-sampling is to create a restore HR (G_{RHR}) image of various sizes from an LR (G_{LR}) image. It refers to the image grid illustrated on the left side of Fig. 1. Let's assume a 3×3 local region of LR image (G_{LR}) that zooms by a factor of two. The image grid is then enlarged and shown on the right side of Fig. 1. The grid is made up of known pixels (in black) and unknown pixels (white circle). Image interpolation is a technique for determining unknown pixel values using known pixels. The weighted average of neighboring pixels is typically used to predict the missing pixel value, which introduces error. As a result, artifacts such as blurring, ringing, and blocking appear in the interpolated image. This necessitates the development of a new interpolation method capable of restoring all edge and texture details in an HR image while avoiding artifacts.

4. Proposed Method

As shown in Fig. 2, the proposed technique relies on optimized pre-processing and edge-preserving spline (e-spline). To produce the adaptively blurred image, the LR image is first passed through the LA gaussian filter, whose center pixels are varied according to the local region variance. This adaptive blurring is used to highlight the image's HF and deemphasize the image's smooth region of image. The LA filtered image is then combined with the LR image using a CS optimized scale factor in the following step. The sharpened LR image is then interpolated with the B-spline method, followed by edge expansion. Edge expansion is used to reduce the zigzag artifacts from the interpolated image edge. Finally, as shown in Fig. 2, this method produces a more natural-looking interpolated image.

4.1. LA filtering

The LA filtering structure is based on the USM concept. The sharpened image is formed by combining the original image with the HPF image during the USM operation. The LR G_{LR} image is extracted from the captured image G_{CHR} by deleting the row and column. As illustrated in Fig. 2, the LR image G_{LR} is used as an input image. Equ. (1) defines the USM concept as follows:

$$G_{SLR} = G_{LR} + kH(x,y) \tag{1}$$

where G_{SLR} denotes the sharpened LR image following a USM operation. (x,y) represents the pixel coordinate of image. The scaling or gain factor is denoted by k. And $H(x,y)$ is the HPF image produced by

$$H(x,y) = k_h(x,y) \otimes G_{LR} \tag{2}$$

where $\otimes$ denotes the convolution operation and k_h is the high pass kernel. Convolving k_h with G_{LR} yields the $H(x,y)$. In contrast, $H(x,y)$ can be obtained by subtracting the smooth or low pass version of input from G_{LR}. It is also specified as

$$H(x,y) = G_{LR}(x,y) - k_l(x,y) \otimes G_{LR}(x,y) \tag{3}$$

$$H(x,y) = G_{LR}(x,y) - H_{Ab}(x,y) \tag{4}$$

where $k_l(x,y)$ is the low pass kernel and H_{Ab} is the adaptive blur image obtained with algorithm 1.

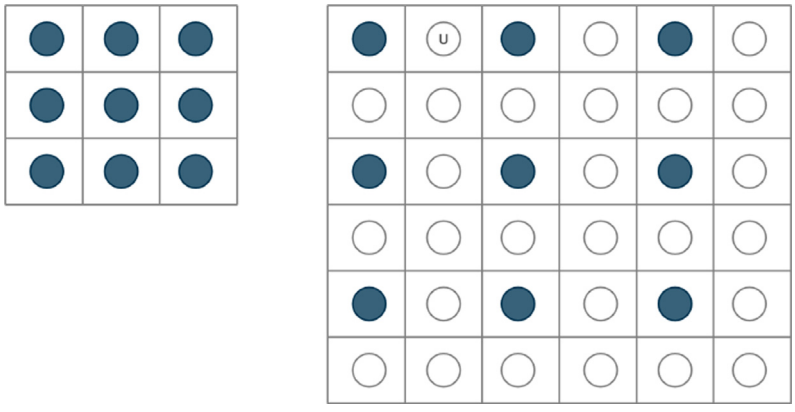

Fig. 1. Image upscaling.

Algorithm 1 LA filtering algorithm

Data: Take the LR image G_{LR}

Result: Adaptive blur image H_{Ab}

Find out the minimum and maximum variance as V_{max} and V_{min} of LR image.

Take the difference $D = V_{max} - V_{min}$.

Divide the variance difference into six no. of equal steps like $S = \frac{V_{max}-V_{min}}{6}$.

for $x = 1 : M$

for $y = 1 : N$

Select the 3×3 local region from G_{LR} and find out the local mean m and variance v.

$m = \frac{1}{9}\sum_{i=-1}^{1}\sum_{j=-1}^{1}\left(G_{LR}\left(x+i,y+j\right)\right)$

$v = \frac{1}{9}\sum_{i=-1}^{1}\sum_{j=-1}^{1}\left(G_{LR}\left(x+i,y+j\right)-m\right)^2$

if $v > V_{max} - S$ **then**
| $C_p = 2$

else if $v > V_{max} - 2S$ *and* $v \leq V_{max} - S$ **then**
| $C_p = 3$

else if $v > V_{max} - 3S$ *and* $v \leq V_{max} - 2S$ **then**
| $C_p = 4$

else if $v > V_{max} - 4S$ *and* $v \leq V_{max} - 3S$ **then**
| $C_p = 8$

else if $v > V_{max} - 5S$ *and* $v \leq V_{max} - 4S$ **then**
| $C_p = 16$

else
| $C_p = 32$

end

The LA gaussian kernel is $g_f = \frac{1}{C_p+12}\begin{vmatrix} 1 & 2 & 1 \\ 1 & C_p & 1 \\ 1 & 2 & 1 \end{vmatrix}$

The adaptive blurred image is $H_{Ab} = \frac{1}{C_p+12}\sum_{i=-1}^{1}\sum_{j=-1}^{1} g_f\left(i,j\right) G_{LR}\left(x+i,y+j\right)$

end

end

Blur is introduced in interpolation as a result of the incorrect estimation of pixel value at an unknown location. To address this issue, the USM concept [26] is applied prior to up-sampling, resulting in less blurring after upscaling. The low pass or smoothed version of the image is extracted using LA gaussian filtering, denoted as g_f. Over the G_{LR} image, the gaussian mask g_f is applied. The variance of the local 3×3 regions of G_{LR} affects the center pixel C_p value of the gaussian masks g_f. Because our goal is to make high variance regions or image edges more pronounced. The statistical variance is highest in the local region containing the image edges. If the local variance is high, the gaussian filter's center pixel weight will be reduced, resulting in more blurring. As mentioned in algorithm 1, the weight of the center pixel is increased if the local region variance is low in order to obtain a less smooth image. Finally, deducts the local adaptive blurred H_{Ab} image from the G_{LR} and extracts the sharpened HPF $H(x,y)$ image as given in Equ. (3). The $H(x,y)$ is scaled and added to the G_{LR} using a gain factor k. As a result, the sharpened LR image G_{SLR} obtained in Equ. (1).

Because the scale factor k influences G_{SLR}, it is optimized using the CS algorithm. The gain factor indicates how much of $H(x,y)$ is contributed for detailed and sharpened G_{SLR}. Meta-heuristic optimization methods such as particle swarm optimization (PSO) [27], CS [28], and genetic algorithm (GA) [29] are used in single or multiple parameter optimization. Because the CS algorithm requires only one parameter, population, to be initialized at the start of the method, it is used for optimizing the gain k factor.

4.2. CS Optimization

The CS [30] is inspired by cuckoo species' obligate brood parasitism, which involves laying eggs in host bird nests. It is preferred over PSO because PSO [31] demanded that motion inertia and coefficients be assumed at the start of the algorithm. If the initial parameter assumption is incorrect, the method is not expected to converge. When compared to GA [32], CS is much simpler because it does not include mutation, crossover, or selection stages.

Each egg in a nest represents a solution in computer science, and a cuckoo egg represents a new solution. The goal is to use new and potentially better solutions (cuckoos) to replace less-than-ideal solutions in nests. Each nest contains one egg, which represents one solution, or multiple eggs, which represent multiple solutions. The CS is based on three fundamental rules.

- Each cuckoo lays one egg at a time and deposits it in a randomly selected nest.
- The best nests with high quality eggs (solution) will carry over to the next generation.
- The number of host nests is fixed, and a host has a chance of discovering an alien egg with probability $p \in [0,1]$.

The host bird can either discard the egg or abandon the nest to build a new nest in a different location. The characteristic equation is as follows:

$$c(t+1) = c(t) + s.l(\beta) \tag{5}$$

where $c(t)$ represents the current position. The next position is represented by the $c(t+1)$. $l(\beta)$ is derived from the Levy distribution, where Levy flight represents the probability of a transition. The s represents the step size. The scale factor required to optimize is encoded in the cuckoo egg in CS. In this, is only one gain factor k is to optimize. There are n nests in the CS, i.e.$x_i, i = 1,2,3,...,n$. The nest is initially assigned a random position within the solution space. Then, at random, select a cuckoo from a specific nest and replace its solutions with the Levy flight algorithm described in algorithm 2. The best nest or solution is then chosen based on the fitness function and moved towards the optimal solution according to generation $t = 1,2,3,\ldots,g$.

Algorithm 2 CS algorithms

Result: Suitable value of k

Objective function $f(x)$, $x = (x_1, x_2, x_3, ..., x_d)$.

Take the HPF image $H(x,y)$ and LR image G_{LR}. Generate an initial population of n host nests.

while $t < g$ **do**

Get a cuckoo randomly $(say\ i)$ and replace its solutions by performing Levy flight.

Evaluate its fitness f_i. For maximization $f_i \propto f(x_i)$. Choose a nest among $n(say\ j)$ randomly.

if $f_i > f_j$ **then**

Replace j with a new solution.

else

Keep j as a solution.

end

end

4.3. Levy flight

In between consecutive jumps, Levy flight takes a random walk. In Equ. (5), a power law step length with Mantegna's distribution is used to generate random steps given.

$$l(\beta) = t^{-\beta},\ 1 < \beta < 3$$

The distance covered in a fixed number of iterations is determined by the step length β. If β is too small, the new solution will be trivial, and if β is too large, the solution will be out of the search space. The step size s is represented as

$$s = 0.01 * \left(\frac{m_i}{n_i}\right)^{\frac{1}{\beta}}$$

where m and n are normally distributed variables i.e. $m_i = N(0, \sigma_m^2)$ and $n_i = N(0, \sigma_n^2)$ where σ_m and σ_n are defined as

$$\sigma_m = \left(\frac{sin(\frac{\pi * \beta}{2}).\Gamma(1+\beta)}{\Gamma(\frac{1+\beta}{2}).\beta.2^{\frac{\beta-1}{2}}} \right)$$

and

$$\sigma_n = 1$$

4.4. B-spline interpolation

Image up-sampling deals with fitting a continuous curve over the discrete point using known information. In the case of image up-sampling, the missing pixels in the image grid are filled up using the known pixel information. For smoothing images, the cubic B-spline [33] up-sampling method outperforms the traditional Bilinear, Bicubic, and Lanczos interpolation methods. A section of the polynomial curve that passes through the knot represents the cubic B-spline [34], which is a degree 3 polynomial. With the curvature property, the polynomial curve produces a better prediction of unknown pixel values. As a result, it produces minimal optimum quality up-sampling. Equ. (6) can be used to perform 2D interpolation over a sharpened LR G_{SLR} image of size $M \times N$.

$$G_{HR}(x,y) = \sum_{i=1}^{M-1}\sum_{j=1}^{N-1} C(i,j)\beta^3(x-i)\beta^3(y-j) \tag{6}$$

where $x, y \in R$ and $i, j \in z$. The cubic B-spline coefficient is described by $C(i,j)$. $\beta^3(x)$ represents the third order cubic B-spline convolution kernel. $\beta^3(x)$ is obtained by convolution of a basic B-spline kernel of degree 0 four times, i.e. $\beta^0(x)$. The formulary for calculating $\beta^3(x)$ is interpreted as follows:

$$\beta^3 = \begin{cases} \frac{2}{3} - \frac{1}{2}|x|^2(2-|x|) & 0 \leqslant |x| < 1 \\ \frac{1}{6}(2-|x|)^3 & 1 \leqslant x < 2 \\ 0 & otherwise \end{cases} \tag{7}$$

$$\beta^0(x) = \begin{cases} 1 & -0.5 \leqslant |x| < 0.5 \\ \frac{1}{2} & |x| = 0.5 \\ 0 & otherwise \end{cases} \tag{8}$$

Because of the diminishing property [35], this is the smoothest interpolating function. So preserving the edge part of the interpolated image motivates us to improve the image's high variance or edge part. In addition, to remove the zigzag artifacts and obtain a crisper interpolated image, the expansion after interpolation is performed. To detect the edge, a canny edge detection method is used before an edge-based expansion method. The expansion is applied after interpolation to reduce artifacts caused by up-sampling such as blurring, zigzag, and blocking. Because the edges are blurred in the non-stationary region of the image, the edge detection operator is critical. The pixel value of the edge pixels changes quickly in nonstationary regions. Because of this, image interpolation is not well approximated, and blur is introduced. It is required for the detection of a true edge from an up-sampled image using a canny-based edge detection technique. The canny edge detector is used for the following facts.

- It will detect a one-pixel wide edge that is close to being a true or real edge.
- When the signal-to-noise ratio is maximized, the possibility of false edge detection is very low.

Table 1
3×3 matrix taken from interpolated image.

E(x-1, y-1)	E(x-1, y)	E(x-1, y + 1)
E(x, y-1)	E(x, y)	E(x, y + 1)
E(x + 1, y-1)	E(x + 1, y)	E(x + 1, y + 1)

4.5. e-spline algorithm

Initially, the LR image is expanded using the B-spline interpolation technique. The canny edge detection algorithm is then applied to an interpolated image. In the third stage, values will change based on the orientation and detected edge of the neighboring edge pixel, as described in Equ. (11) and (12). Following the completion of the preceding steps, the resulting image is our true edge-preserving spline, i.e. e-spline image. Let $E(x,y)$ be the edge pixel detected by the canny edge detector shown in the following Table 1. According to this, the horizontal and vertical direction changes at (x,y) pixel can be expressed as

$$\begin{aligned} SD_h(x,y) = &\frac{1}{4}[E(x-1,y-1) - E(x-1,y+1)] \\ &+ \frac{1}{2}[E(x,y-1) - E(x,y+1)] \\ &+ \frac{1}{4}[E(x+1,y-1) - E(x+1,y+1)] \end{aligned} \tag{9}$$

$$\begin{aligned} SD_v(x,y) = &\frac{1}{4}[E(x-1,y-1) - E(x+1,y-1)] \\ &+ \frac{1}{2}[E(x-1,y) - E(x+1,y)] \\ &+ \frac{1}{4}[E(x-1,y+1) - E(x+1,y+1)] \end{aligned} \tag{10}$$

If $|SD_h(x,y)| \geqslant |SD_v(x,y)|$ then it indicates edge is in the vertical direction so change the adjacent pixel values of the edge detected pixel (x,y) by

$$E(x,y-1) = \frac{1}{2}(E(x,y-1) + E(x,y-2)) \tag{11}$$

$$E(x,y+1) = \frac{1}{2}(E(x,y+1) + E(x,y+2)) \tag{12}$$

If $|SD_h(x,y)| \leqslant |SD_v(x,y)|$ then it represents the horizontal changes occurs in image. The adjacent pixels will be changed as given in

$$E(x+1,y) = \frac{1}{2}(E(x+1,y) + E(x+2,y)) \tag{13}$$

$$E(x-1,y) = \frac{1}{2}(E(x-1,y) + E(x-2,y)) \tag{14}$$

Pixels $E(x+2,y)$ and $E(x-2,y)$ are located on the lower and upper sides of pixels $E(x+1,y)$ and $E(x-1,y)$, respectively. The pixel between two edge pixels remains unchanged. Following this edge expansion, the edges of the HR image G_e are aggregated with G_{HR} to form G_{RHR}, and the up-sampled image appears more natural-looking. Several artifacts are being reduced as a result of edge expansion and interpolation.

$$G_{RHR} = G_{HR} + G_e \tag{15}$$

5. Experimental Result

Various images from standard image processing, Set 5, and Set 14 are used to evaluate the performance of the proposed method.

Any scaling factor is used to downsample the captured images. In this case, the proposed method is used to upscale the LR image and recover the HR image from it. The proposed technique can be objectively evaluated using PSNR, structural similarity index (SSIM) [36], and feature structural index measurement (FSIM). Different upscaling factors, such as 2 and 4, are used to assess the performance of the proposed scheme. PSNR is the signal power divided by the noise power. If its value is high, it indicates that the signal has been restored successfully. The PSNR is calulated by

$$PSNR = 10.log_{10}\frac{MAX}{MSE} \tag{16}$$

where MAX represents the maximum intensity value of pixel of image and MSE is the mean sqared error and determined by

$$MSE = \frac{1}{M \times N}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[G_{CHR}(x,y) - G_{RHR}(x,y)]^2 \tag{17}$$

The structural similarity between the upscaled image and the captured image is represented by the SSIM. The SSIM is obtained by using

$$SSIM(k,l) = \frac{(2\mu_k\mu_l + v_1)(2\sigma_{kl} + v_2)}{(\mu_k^2 + \mu_l^2 + v_1)(\sigma_k^2 + \sigma_l^2 + v_2)} \tag{18}$$

where k and l be the local window of size $W \times W$ of up-sampled and captured image. μ_k and μ_l is the average value of local window k and l. σ_k^2 and σ_l^2 is the variance of window k and l. σ_{kl} is the local window covariance. v_1 and v_2 is the variable use to stabilize the Equ. (18) when zero appear in denominator.

$$v_1 = (x_1D)^2 \text{ and } v_2 = (x_2D)^2$$
$$\text{where the default value of } x_1 = 0.01 \text{ and } x_2 = 0.03. \tag{19}$$

where D is the dynamic range of image. The performance tables demonstrate the proposed technique adaptability to different image sizes. Face, Airplane, and Baby images are considered for subjective evaluation because they belong to high (edge or high variance region), medium (medium variance region), and low (smooth or less variance region) frequency bands. The error image, which is the difference between the interpolated and original images, is used to evaluate the proposed method's performance. If the error difference is small, it demonstrates better edge and texture restoration; however, if the difference is visible, edge preservation will not occur. Apart from the state-of-the-art algorithms, the proposed method outperforms the Lanczos and other state-of-the-art methods for various images, as shown in Tables 2 and 3. Table 4 shows the average PSNR gain of various cutting-edge techniques. The experimental results of the OLA e-spline, as well as existing algorithms, are shown below. Fig. 3 and 4 depict the subjective performances of Baby (512×512) and Face (276×276) using algorithms at 1:4 upscaling.

Table 2
PSNR (dB) outcomes of several 1:4 upscaling method

Method Image	Performance Parameter	Bilinear [1]	Bicubic [2]	Lanczos [4]	NEDI [5]	ICBI [15]	DST [9]	Edge error [10]	USM	Proposed
Monkey	PSNR (dB)	24.4462	23.8962	29.7933	23.0902	23.2438	23.2841	22.5778	23.0576	**30.1449**
	SSIM	0.5760	0.6192	0.5708	0.5265	0.5412	0.6384	0.5608	0.6103	**0.6471**
	FSIM	0.9290	0.9590	0.8704	0.9489	0.9508	0.9517	0.9949	0.9449	**0.9991**
Airplane	PSNR (dB)	32.3601	31.2664	39.6672	29.8161	30.1243	30.2296	28.3969	29.8647	**44.8822**
	SSIM	0.9209	0.7483	0.8534	0.6881	0.7025	0.9047	0.8778	0.8964	**0.9284**
	FSIM	0.9843	0.9625	0.9326	0.9598	0.9611	0.9612	**0.9972**	0.9596	0.9722
House	PSNR (dB)	32.0809	31.5451	32.8524	30.6702	30.5516	30.6328	29.6852	30.4241	**33.0807**
	SSIM	**0.8508**	0.5548	0.6041	0.5119	0.5208	0.8404	0.7949	0.8327	0.6457
	FSIM	0.8933	0.8824	0.7512	0.8848	0.8862	0.8866	**0.9749**	0.8780	0.7394
Peppers	PSNR (dB)	31.7061	31.1493	37.6293	30.4879	30.5189	30.5594	29.4785	30.3161	**38.7196**
	SSIM	0.7721	0.5500	0.6739	0.5036	0.5148	0.7852	0.7586	0.7765	**0.7977**
	FSIM	0.9872	0.9741	0.9387	0.9744	0.9020	0.9750	**0.9973**	0.9723	0.9191
Lena	PSNR (dB)	32.8712	31.6746	35.6507	30.6258	30.7563	30.8307	30.5510	30.6464	**39.8594**
	SSIM	0.8733	0.6223	0.7349	0.5672	0.5801	0.8634	0.8301	0.8551	**0.8870**
	FSIM	0.9876	0.9635	0.9374	0.9664	0.9675	0.9680	**0.9975**	0.9671	0.9404
Butterfly	PSNR (dB)	24.6972	24.5919	32.3331	23.6608	23.9255	23.7990	22.6180	23.2533	**38.8828**
	SSIM	0.8919	0.7695	0.8629	0.7730	0.7719	0.8840	0.8284	0.8710	**0.9112**
	FSIM	0.8866	0.8851	0.8840	0.8914	0.8867	0.8823	**0.9907**	0.8683	0.9073
Face	PSNR (dB)	29.6793	29.4917	31.0265	28.6688	28.6788	28.7375	28.5490	28.4365	**34.1634**
	SSIM	**0.7437**	0.5964	0.6240	0.5440	0.5604	0.7372	0.6972	0.7271	0.6828
	FSIM	0.8782	0.9042	0.8840	0.8762	0.8845	0.8860	0.9749	0.8732	**0.9798**
Foreman	PSNR (dB)	31.5793	31.5080	36.3011	30.4322	31.6332	31.5736	27.8672	30.9274	**42.5127**
	SSIM	0.9219	0.8145	0.9039	0.7590	0.7667	0.9048	0.8863	0.8953	**0.9317**
	FSIM	0.9322	0.9401	0.9313	0.9264	0.9273	0.9256	0.9292	0.9169	**0.9555**
Baby	PSNR (dB)	31.3357	32.6657	35.7413	31.4252	31.7574	31.8357	31.0620	31.5253	**41.6809**
	SSIM	0.7804	0.8045	0.9047	0.7557	0.7707	0.9225	0.8855	0.9143	**0.9455**
	FSIM	0.8002	0.9682	0.9703	0.9716	0.9703	0.9740	**0.9955**	0.9718	0.9656
Bird	PSNR (dB)	32.7389	30.9862	34.6196	29.6286	30.0144	30.0639	29.7283	29.6890	**41.0243**
	SSIM	0.9526	0.8757	0.9208	0.8613	0.8682	0.9326	0.9120	0.9256	**0.9682**
	FSIM	0.9411	0.9496	0.9331	0.9361	0.9405	0.9409	0.9619	0.9361	**0.9709**
Coast guard	PSNR (dB)	27.0121	27.6962	29.9930	25.9980	26.4020	26.5038	25.3433	26.1219	**30.3749**
	SSIM	**0.7247**	0.6919	0.5833	0.5694	0.6022	0.7196	0.6131	0.6912	0.6847
	FSIM	0.8157	0.8603	0.7593	0.8163	0.8357	0.6912	**0.9581**	0.8179	0.8038
Barbara	PSNR (dB)	25.6966	25.6844	31.3104	24.5322	24.7951	24.9268	24.7039	24.6878	**35.6204**
	SSIM	0.7892	0.6873	0.7019	0.6121	0.6232	0.7615	0.7407	0.7615	**0.7977**
	FSIM	0.9692	0.9586	0.9007	0.9492	0.9519	0.9525	0.9675	0.9465	**0.9975**
Woman	PSNR (dB)	29.2734	28.2837	34.5242	26.9773	27.5344	27.6147	27.0982	27.1863	**37.2659**
	SSIM	0.9209	0.8343	0.9014	0.8124	0.8192	0.9098	0.8773	0.8986	**0.9441**
	FSIM	0.9267	0.9312	0.9182	0.9147	0.9188	0.9199	0.9207	0.9122	**0.9572**
Fence	PSNR (dB)	22.5609	23.2760	30.9314	21.5715	21.6294	21.7854	21.0046	21.4708	**32.8789**
	SSIM	0.7259	**0.8343**	0.7123	0.5549	0.5673	0.7235	0.6162	0.6990	0.7807
	FSIM	0.8050	0.8371	0.8174	0.8007	0.8012	0.8001	**0.9627**	0.7872	0.8941

Table 3
PSNR (dB) results of the 1:16 upscaling algorithms

Method Figure	Performance Parameter	Bilinear [1]	Bicubic [2]	Lanczos [4]	NEDI [5]	ICBI [15]	DST [9]	Edge error [10]	USM	Proposed
Monkey	PSNR (dB)	20.5007	19.8446	28.9460	19.1560	19.4593	19.5353	19.2674	19.5602	**29.2381**
	SSIM	0.3171	0.1164	0.2625	0.1642	0.1909	0.2888	0.2781	0.2868	**0.3194**
	FSIM	0.6787	0.7269	0.7504	0.6404	0.6929	0.6950	0.7634	0.6669	**0.7983**
Airplane	PSNR (dB)	22.9181	21.0856	33.4825	19.9527	20.2535	20.4264	19.7982	20.5224	**35.4010**
	SSIM	0.6763	0.1900	0.5882	0.1165	0.1475	0.6311	0.6052	0.6319	**0.6817**
	FSIM	0.7701	0.7516	0.7200	0.7330	0.7456	0.7426	0.8073	0.7345	**0.8126**
House	PSNR (dB)	24.3337	23.1145	34.2867	20.9846	22.5184	22.4629	20.8661	22.5777	**34.9268**
	SSIM	0.6463	0.1038	0.5760	0.1024	0.2821	0.6136	0.5604	0.6141	**0.6817**
	FSIM	0.7005	0.6723	0.6610	0.6700	0.6893	0.6896	**0.8115**	0.6798	0.7470
Peppers	PSNR (dB)	25.5633	22.8124	33.6120	20.4178	22.2461	22.2855	21.8739	22.3811	**35.5670**
	SSIM	**0.6180**	0.1871	0.5178	0.2085	0.1621	0.5772	0.5610	0.5765	0.5983
	FSIM	0.8354	0.8153	0.7441	0.7784	0.8054	0.8050	0.8201	0.7991	**0.8393**
Lena	PSNR (dB)	25.0018	21.9761	32.4643	20.7357	21.1333	21.2539	21.3502	21.4767	**34.1476**
	SSIM	0.6731	0.1849	0.5697	0.1638	0.1494	0.6158	0.5967	0.6184	**0.7573**
	FSIM	0.8322	0.7737	0.6552	0.7663	0.7784	0.8054	0.8249	0.7751	**0.8448**
Butterfly	PSNR (dB)	24.4592	24.0965	29.7151	27.6529	27.8397	28.7500	26.8436	27.5621	**31.2091**
	SSIM	0.4468	0.5473	0.6744	0. 6995	0.5899	0.6322	0.5355	0.6187	**0.7056**
	FSIM	0.6307	0.6025	0.8562	0.7089	0.7260	0.7129	0.8314	0.7108	**0.9072**
Face	PSNR (dB)	24.9707	22.6436	28.8847	23.8705	22.0217	22.1316	22.5800	22.1106	**29.9640**
	SSIM	**0.5126**	0.2734	0.3823	0.2827	0.2586	0.4608	0.4652	0.4855	0.4377
	FSIM	0.6786	0.6897	0.7622	0.7403	0.6753	0.6761	**0.8320**	0.6310	0.7418
Foreman	PSNR (dB)	23.2684	21.0374	30.7249	21.3748	21.8225	21.7561	22.5066	21.7035	**31.5629**
	SSIM	0.6796	0.3003	0.6069	0.5569	0.5600	0.6302	**0.7319**	0.6288	0.7122
	FSIM	0.7593	0.7401	0.7314	0.7443	0.7260	0.7440	**0.8638**	0.7384	0.7930
Baby	PSNR (dB)	24.6613	21.9550	31.4293	21.8210	21.1890	21.2842	20.9756	21.3870	**32.1540**
	SSIM	0.6840	0.6245	0.6108	0.2248	0.4873	0.6375	0.6016	0.6378	**0.6871**
	FSIM	0.8016	0.7753	0.8013	0.7746	0.7680	0.6899	**0.9245**	0.7642	0.8466
Bird	PSNR (dB)	21.8579	19.7687	29.4002	19.8209	21.7081	22.7582	23.9374	22.8935	**30.8683**
	SSIM	0.6076	0.3032	0.6592	0.3189	0.5483	0.6177	0.6849	0.6121	**0.7195**
	FSIM	0.7434	0.7472	0.8343	0.7190	0.7210	0.7219	**0.8725**	0.8252	0.7862
Coast guard	PSNR (dB)	22.2172	21.7288	29.2497	21.5590	21.6726	21.4775	20.2663	21.4684	**29.6861**
	SSIM	0.3624	0.2792	**0.4039**	0.3596	0.2553	0.3275	0.3294	0.3266	0.3526
	FSIM	0.4842	0.5134	0.6004	0.4850	0.4867	0.4825	**0.7876**	0.4670	0.6117
Barbara	PSNR (dB)	21.1621	22.6183	29.6559	20.0106	21.3644	21.2005	21.7715	22.4757	**30.3396**
	SSIM	0.5068	0.3765	0.6102	0.1604	0.4474	0.4449	0.3472	0.5548	**0.6744**
	FSIM	0.6610	0.7495	0.7133	0.7127	0.6870	0.7161	**0.9151**	0.7099	0.7929
Woman	PSNR (dB)	19.5723	20.7600	30.5620	21.1021	21.6621	20.7653	21.3440	21.7487	**31.0933**
	SSIM	0.6672	0.4667	0.6811	0.4839	0.4940	0.5622	0.5620	**0.7163**	0.6732
	FSIM	0.7564	0.7110	0.6409	0.7921	0.6011	0.7633	0.6689	0.7550	**0.7803**
Fence	PSNR (dB)	19.1610	20.9966	29.7750	21.2938	20.3229	21.3301	22.3603	24.3371	**30.4959**
	SSIM	0.3761	0.3465	0.6174	0.3984	0.2119	0.4434	0.4128	0.5337	**0.6803**
	FSIM	0.6304	0.6090	0.7967	0.5874	0.5966	0.8466	0.7401	0.7888	**0.8967**

Table 4
Avg. PSNR gain of different upscaling algorithm

Method Scale factor	Bilinear	Bicubic	Lanczos	NEDI	ICBI	DST	Edge error	USM	Proposed
×2	29.1455	28.8368	33.7454	25.7827	26.0014	28.0269	27.0474	27.6862	37.2207
×4	22.8319	21.7455	30.8705	21.4089	21.8009	21.9583	21.8386	22.3003	31.9044

Table 5
Processing time of different upscaling algorithms

Processing Time (sec) for ×2	Bilinear	Bicubic	Lanczos	NEDI	ICBI	DST	Edge error	USM	Proposed
House (256×256)	0.0098	0.3110	1.9308	30.3063	0.3030	0.6235	1.6804	0.4751	7.2829
Butterfly (256×256)	0.0082	0.3535	1.7877	106.280	0.2674	0.6060	1.6069	0.4690	6.8980
Foreman (288×352)	0.0087	0.4902	2.2775	101.12	0.3861	0.9184	1.6714	0.7545	14.8170
Women (344×228)	0.0090	0.4106	2.4332	103.03	0.2855	0.7894	1.6513	0.5039	9.4480
Fence (256×256)	0.0102	0.3616	1.8917	88.8374	0.3132	0.5721	1.6830	0.5128	7.2449

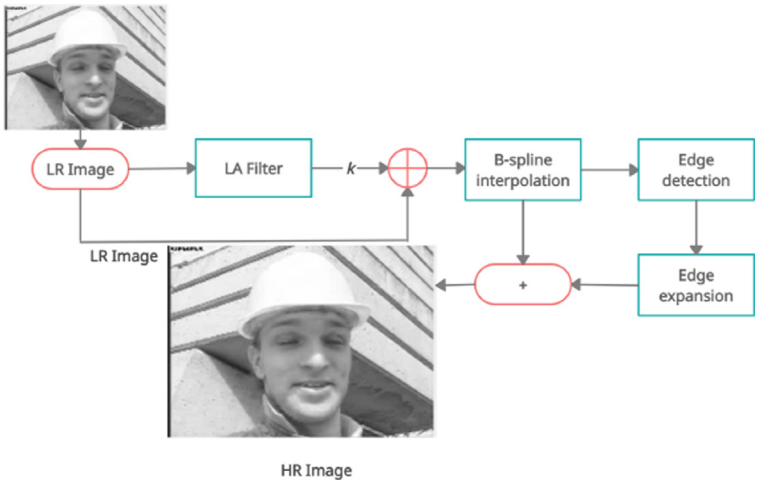

Fig. 2. Block diagram of proposed method.

Fig. 5 depicts HF region (Hair of Face image) restoration at higher upscaling. Fig. 6 shows the edge and texture preservation of an airplane image at 1:4 upscaling. The error image of the Airplane image and the Lena image are shown in Fig. 7 and 8, respectively. Tables 2 and 3 show the PSNR, SSIM, and FSIM parameters of various methods, as well as the OLA e-spline, at 1:4, 1:16 upscaling. The peak performance parameters are highlighted. Table 5 shows the execution time of various methods for various image sizes at 1:4 upscaling. Table 4 shows the average PSNR gain of the proposed and other state-of-the-art algorithms at 1:4 upscaling. The proposed scheme outperforms the Bilinear, Bicubic, Lanczos, and USM interpolated methods by 8.0752, 8.3839, 3.4753, and 9.5355 dB, respectively.

5.1. Result Analysis

Face, Baby, and Airplane images are considered for subjective evaluation. Instead of selecting a specific region of the image, such as the low, medium, or high frequency category. Here, the Baby, Airplane, and Face images represent low, medium, and high frequencies, respectively, and have been enlarged as shown in Fig. 3, 6, and 4. The Baby falls into the category of a smooth image, as it contains the majority of the low region variance. For this reason, it is regarded as a low-frequency image. As shown in Fig. 3, homogeneous regions are completely preserved after enlargement 3 (j).

The hair in the Face image represents the image's HF or edges. The hair is completely emphasized and distinguishable after enlargement to compensate for the HF loss caused by interpolation, as shown in Fig. 4 (j). Similarly, the Airplane image is made up of low, medium, and high-frequency regions, which are fairly enhanced after upscaling. The name of the airplane and its edges are visible after interpolation, as shown in Fig. 6 (j), whereas a smoother interpolated image can be found in (b), (c), and (d). As a result of the subjective analysis, it was concluded that the proposed method, when compared to Lanczos interpolation, preserves the loss edge information as well as the minute details after upscaling. However, the restoration of the edge region is dependent on the depth of degradation of the edges. When compared to other existing algorithms, the OLA e-spline produces the best results by restoring all missing details.

For subjective evaluation, the Face, Baby, and Airplane images are considered. Instead of selecting a particular region of the image, which may be low, medium, or HF category. Here Baby, Airplane and Face images represent low, medium, and HF respectively, and then enlarged as shown in Fig. 3, Fig. 6, and 4. The Baby is a category of a smooth image, as most of the region variance is very low for this, it considered as a low-frequency image. It can be noticed that homogeneous regions are completely preserved after enlargement as shown in Fig. 3 (j). The Face image's hair represents HF or edges of the image. After enlargement, the hair is completely emphasized and distinguishable to overcome the HF loss because of interpolation as given in Fig. 4 (j). Similarly, the Airplane image is a combination of low, medium, and HF regions, and these are enhanced fairly, after upscaling. After interpolation, the name of Airplane and its edges are visible as manifested in Fig. 6 (j) whereas more smoothness interpolated image can be found in (b), (c), and (d). Hence from the subjective analysis, it concluded that the proposed method preserves the loss edge information along with the minute details after upscaling compared to Lanczos interpolation. However, the edge region restoration depends upon the depth of degradation of edges. Compared to other existing algorithms, the OLA e-spline provides the most desirable results by restoring all missing details. The proposed method restoration quality is expressed by the error image. Fig. 7 and 8 depict the error image of Airplane and the Lena image, respectively, at $1:4$ upscaling. More error can be seen at the edge parts of the Airplane image in Fig. 7 (f) and (h). This is due to the fact that edges are not restored in DST and USM. In the case of the Lena image, errors exist in both the low and high-frequency parts of the image, as shown in Fig. 8 (a), (d), and (e). As shown in Fig. 8 (c), the Lanczos interpolated image has less error than the existing algorithms. It is also worth noting that the proposed method produces the least amount of error in both cases. According to Tables 2 and 3, the proposed algorithm performs better objectively for various database images.

In the case of the Baby (512×512) image, the OLA e-spline outperforms Lanczos with a maximum PSNR gain of 5.9396 dB at a $1:4$ upscaling ratio. Similarly, the proposed method outperforms the NEDI, Edge error, and USM interpolation methods by 10.2557, 10.6189, and 10.1556 dB, respectively. At $1:4$ upscaling, the OLA e-spline achieves a PSNR gain of 12.5221 and 4.4841 dB over the Bilinear scheme for HF images like Airplane and Face. The proposed method produces better results in images with low and medium frequency bands, such as Baby and Lena. It produces a visible result because the restoration of HF after interpolation is required. Similarly, the proposed method outperforms other state-of-the-art methods in terms of observable parameters such as PSNR, SSIM, and FSIM at $1:16$ upscaling schemes, as shown in

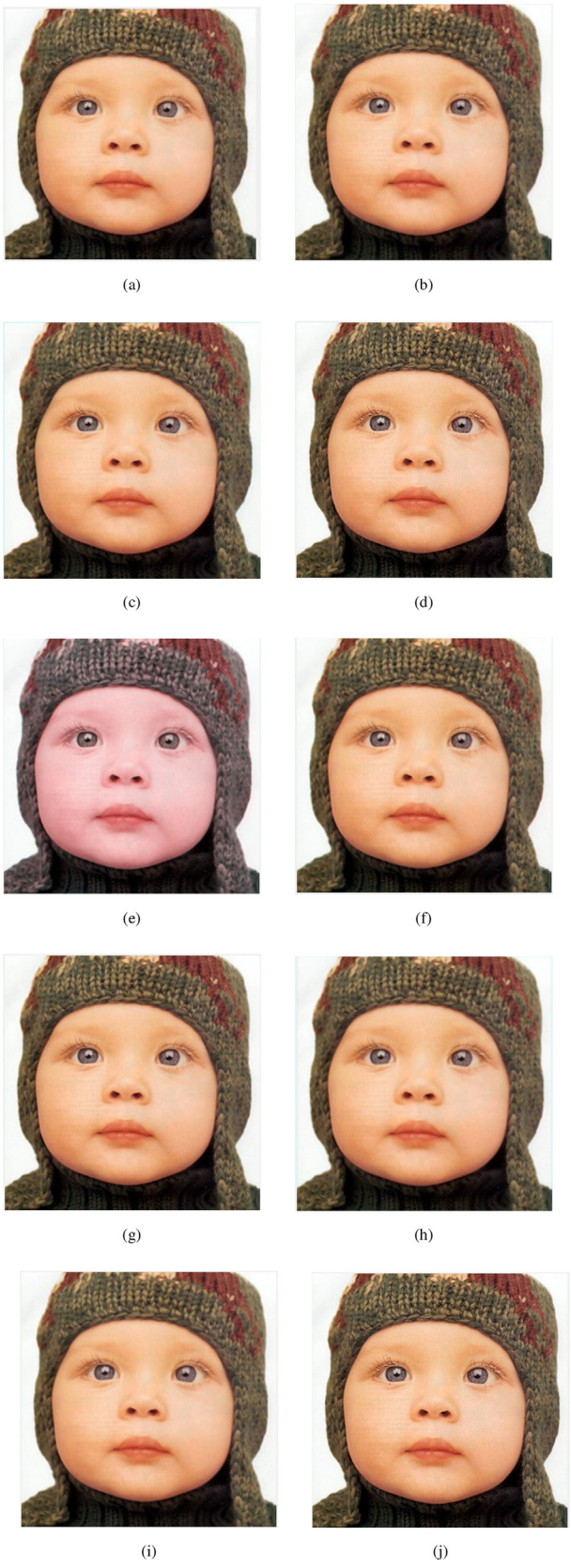

Fig. 3. (a) Baby LR image and results of several upscaling algorithm at 1:4 upscaling by (b) Bilinear [1] (c) Bicubic [2] (d) Lanczos [4] (e) NEDI [5] (f) ICBI [15] (g) DST [9] (h) Edge error [10] (i) USM (j) Proposed method.

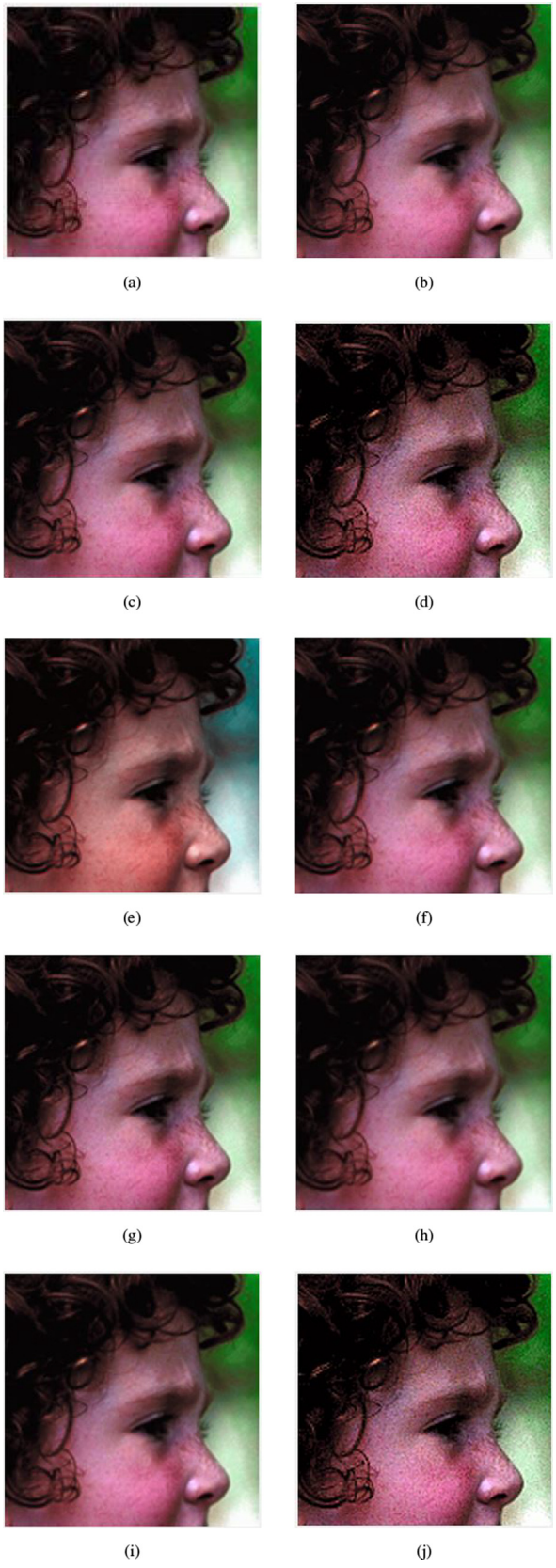

Fig. 4. (a) Face LR image and results of several upscaling algorithm at 1:4 upscaling by (b) Bilinear [1] (c) Bicubic [2] (d) Lanczos [4] (e) NEDI [5] (f) ICBI [15] (g) DST [9] (h) Edge error [10] (i) USM (j) Proposed method.

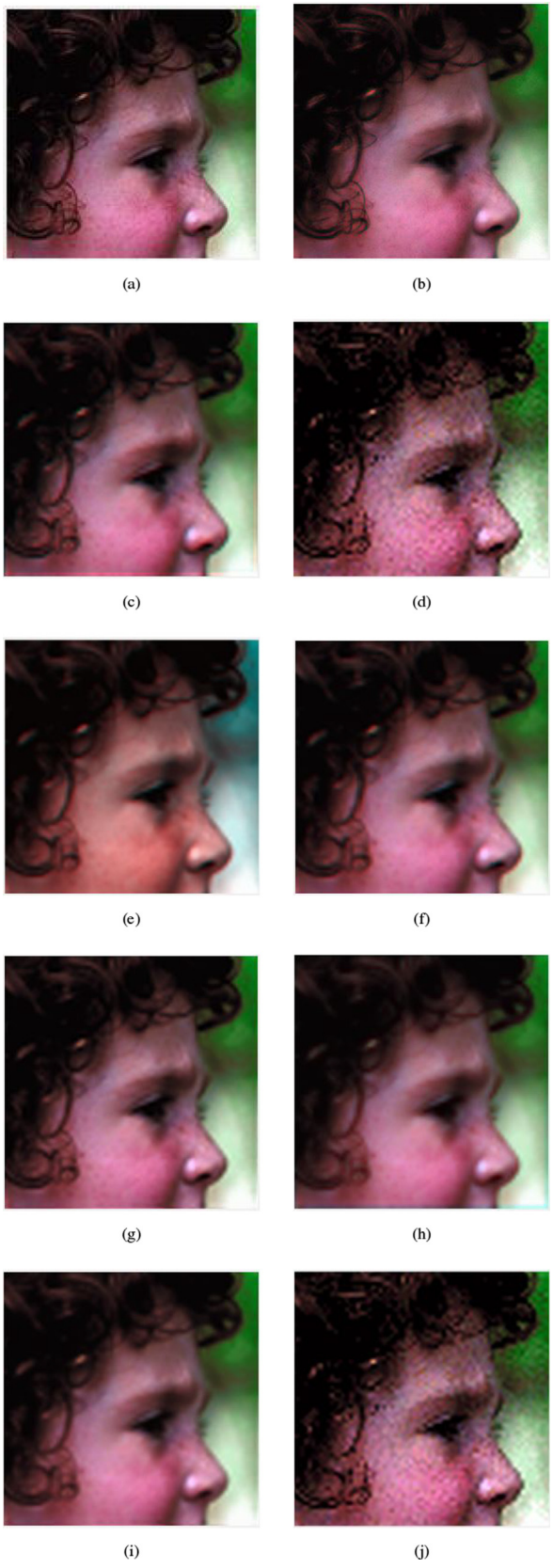

Fig. 5. (a) Face LR image and outcome of different algorithms at 1:16 upscaling by (b) Bilinear [1] (c) Bicubic [2] (d) Lanczos [4] (e) NEDI [5] (f) ICBI [15] (g) DST [9] (h) Edge error [10] (i) USM (j) Proposed method.

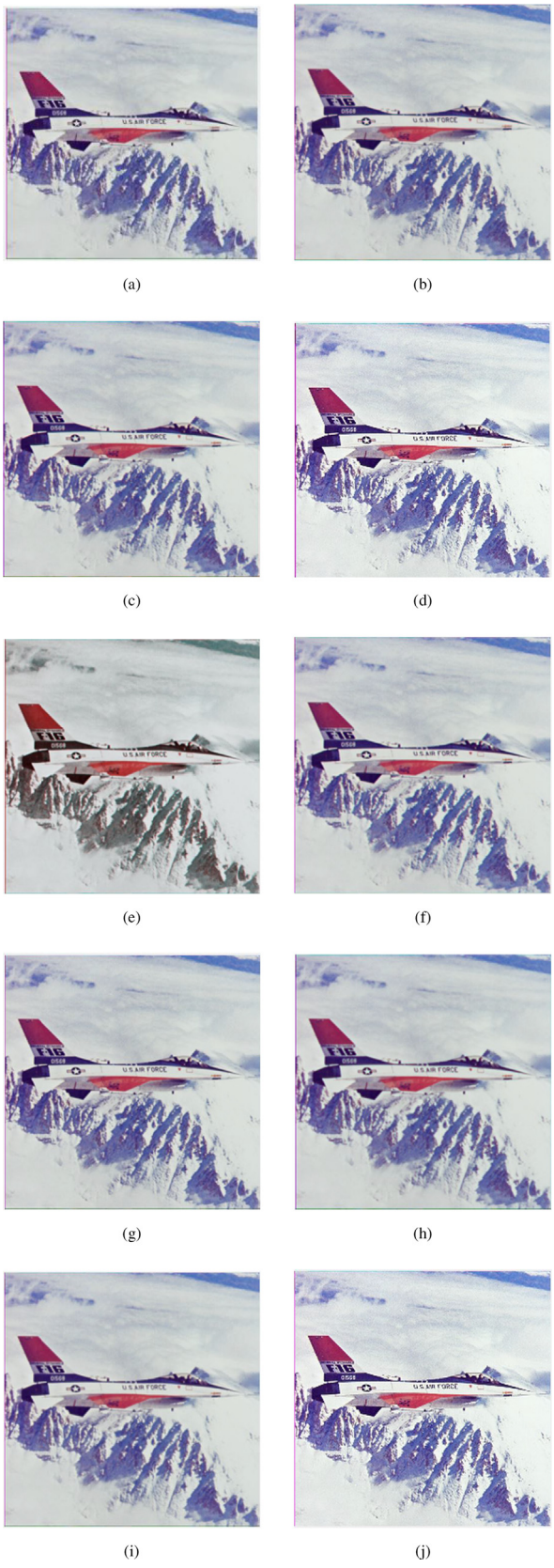

Fig. 6. (a) Airplane LR image and results of several upscaling algorithm at 1:4 upscaling by (b) Bilinear [1] (c) Bicubic [2] (d) Lanczos [4] (e) NEDI [5] (f) ICBI [15] (g) DST [9] (h) Edge error [10] (i) USM (j) Proposed method.

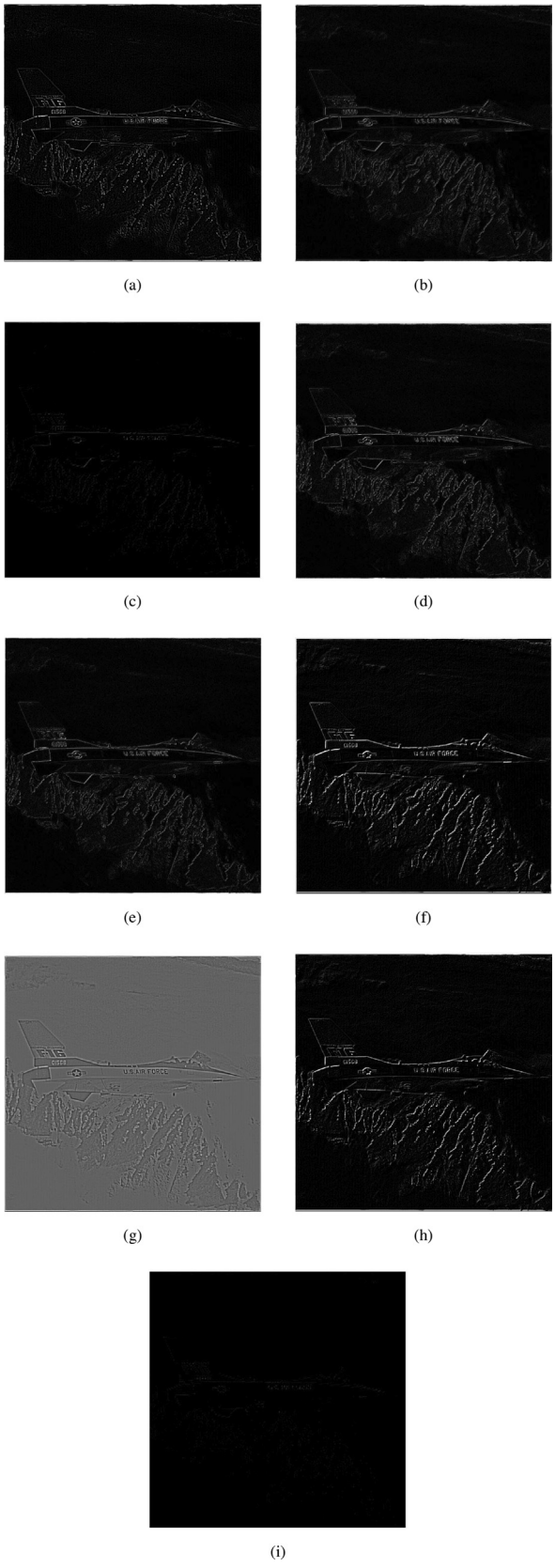

Fig. 7. Airplane error image of several interpolation algorithms at 1:4 upscaling by (a) Bilinear [1] (b) Bicubic [2] (c) Lanczos [4] (d) NEDI [5] (e) ICBI [15] (f) DST [9] (g) Edge error [10] (h) USM (i) Proposed method.

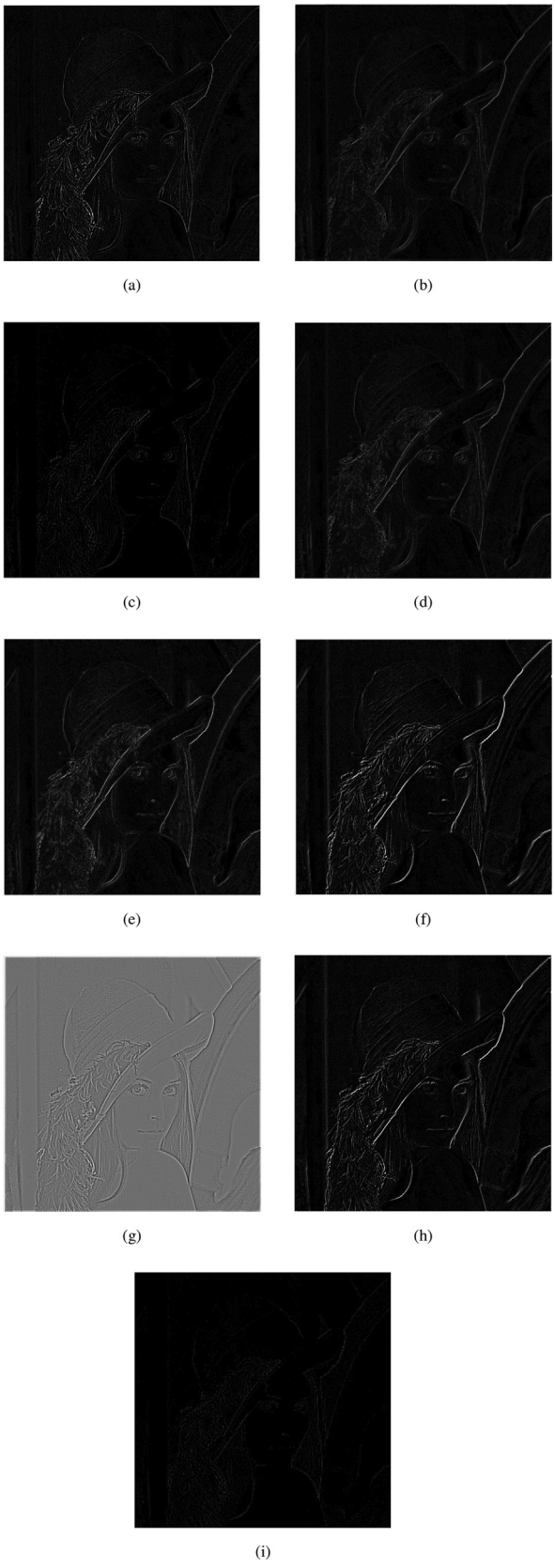

Fig. 8. Lena error images of several interpolation algorithm at 1:4 upscaling by (a) Bilinear [1] (b) Bicubic [2] (c) Lanczos [4] (d) NEDI [5] (e) ICBI [11] (f) DST [9] (g) Edge error [10] (h) USM (i) Proposed method.

Table 3. As a result, the proposed algorithm produces visible results for any upscaling factor of any type of image.

Traditional methods such as Bilinear, Bicubic, and DST perform interpolation without taking smooth (low frequency) and edge (HF) into account separately. Blur is introduced in nonstationary regions as a result. Furthermore, subjective and objective evaluations show that the Lanczos method produces comparable results, but Lanczos interpolation introduces some ringing artifacts in the HR image due to the sinc function. As a result, it will produce poor results when compared to the proposed method. Furthermore, NEDI, and ICBI perform edge interpolation but introduce false edge artifacts in the up-scaled images. In the proposed scheme, the degraded HF of the image is completely stored in the e-spline technique after LA filtering before interpolation and edge expansion after up-sampling. In the USM, the directly HPF version of the LR image is extracted and added to the LR image before up-sampling, resulting in a lower PSNR gain. In addition, the subjective result shows that edge error is superior to USM because it uses the inverse approach to restore the degraded HF. Because the error generated by interpolation is added, this method is capable of restoring the lost edge and fine details due to up-sampling. However, for different scaling-factors, such as 4, its performance is poor, as shown in Table 3. The proposed scheme is significant for using the pre-processing technique before e-spline interpolation; for this, the proposed algorithm produces better results. The USM concept is used for pre-processing. It is used prior to enlargement to restore details lost due to interpolation. As a result, the proposed method is simpler than NEDI, and ICBI. And, if this is implemented in real-time applications, the proposed scheme will be less expensive than the NEDI and ICBI without sacrificing output quality. According to the experimental results, the proposed scheme provides a better HR image both objectively and subjectively when compared to other state-of-the-art methods. However, depending on the processing time, it lags behind the Lanczos method. Because the proposed method employs LA filtering, its execution time is longer than that of Lanczos.

The processing time is the most important factor in a real-time application. However, as shown in Table 5, the proposed method takes longer to execute than USM because it uses adaptive filtering to produce the sharpened HR image. Table 5 shows that the computational time of the suggested method is longer than that of the Bicubic and Bilinear methods. These methods perform uniform interpolation over the image's low and high variance regions. On the other hand, this preprocessing technique results in a more natural-looking HR. Furthermore, NEDI, and ICBI take longer to process than other methods because they perform different interpolation for the image's edge and texture regions.

Based on the objective and subjective experimental results, the OLA e-spline method outperforms the Bicubic, Lanczos, NEDI, and other algorithms discussed in the literature.

6. Conclusion

An experimental result shows that the proposed scheme outperforms the existing schemes both objectively and subjectively. This is due to pre-processing prior to interpolation, followed by edge expansion. As a result, the smooth regions are completely preserved and the medium frequencies are reasonably enhanced. The HF is more focused, allowing it to recover all lost details. The CS optimization is then used to combine the LA blurred image with the LR images with the optimised gain factor, which is used before upscaling to reduce the zigzag and blur artifacts, and the edge expansion of the interpolated image is performed after upscaling. The OLA e-spline is so simple that it can be used in any display device that requires a high resolution image. It will reduce the computational burden on real-time applications. This method adaptively recovers the image's low, medium, and high-frequency regions based on the lost details. According to the results analysis, this elite and sophisticated method performs well for all frequency regions. For scaling factors of 2 and 4, respectively, this method achieves a PSNR gain of 3.4753 and 1.0339 dB over the Lanczos interpolation method. The proposed method achieves a PSNR gain of 9.5345 dB over USM and 9.1938 dB over DST for scaling factor 2. HR image edges are more pronounced and distinguishable from smooth regions. The computational complexity is extremely low. It can be used in consumer electronics applications for this purpose. This work can be expanded by designing an optimized USM filter or by employing a post-processing approach to reduce artifacts caused by upscaling.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Yan F., Zhao S., Venegas-Andraca S.E. Implementing Bilinear Interpolation on Quantum Images, arXiv preprint arXiv:2010.10254 (2020).
[2] Huang Z, Cao L. Bicubic interpolation and extrapolation iteration method for high resolution digital holographic reconstruction. Opt Lasers Eng 2020;130:106090.
[3] Zachariadis O, Teatini A, Satpute N, Gómez-Luna J, Mutlu O, Elle OJ, Olivares J. Accelerating B-spline interpolation on GPUs: Application to medical image registration. Computer Methods Programs Biomedicine 2020;193:105431.
[4] He L, Wu H, Zhao X. The application of Lanczos interpolation in video scaling system based on FPGA. In: Twelfth International Conference on Signal Processing Systems, 11719. International Society for Optics and Photonics; 2021. p. 117190G.
[5] Li X, Orchard MT. New edge-directed interpolation. IEEE Trans Image Processing 2001;10(10):1521–7.
[6] Liu X, Sun J, Yan G, Lu Y, Xue Y, Li G, et al. 32–2: Improvement in Directional Cubic Convolution Image Interpolation. In: SID Symposium Digest of Technical Papers, 51. Wiley Online Library; 2020. p. 455–8.
[7] Zhang L, Wu X. An edge-guided image interpolation algorithm via directional filtering and data fusion. IEEE Trans Image Processing 2006;15(8):2226–38.
[8] Sajjad M, Ejaz N, Mehmood I, Baik SW. Digital image super-resolution using adaptive interpolation based on Gaussian function. Multimedia Tools Appl 2015;74(20):8961–77.
[9] Panda J, Meher S. A Novel Approach of Image Interpolation using DST. In: 2019 Fifth International Conference on Image Information Processing (ICIIP). IEEE; 2019. p. 606–11.
[10] Panda J, Meher S. An Efficient Image Interpolation Using Edge-error Based Sharpening. In: 2020 IEEE 17th India Council International Conference (INDICON). IEEE; 2020. p. 1–6.
[11] Giachetti A, Asuni N. Real-time artifact-free image upscaling. IEEE Trans Image Process 2011;20(10):2760–8.
[12] Rukundo O. Normalized Weighting Schemes for Image Interpolation Algorithms, arXiv preprint arXiv:2011.08559 (2020).
[13] Zheng J, Song W, Wu Y, Liu F. Image interpolation with adaptive k-nearest neighbours search and random non-linear regression. IET Image Proc 2020;14 (8):1539–48.
[14] Zhang J. Weighted-encoding-based image interpolation with the nonlocal linear regression model. Appl Opt 2020;59(28):8588–94.
[15] Byongsu H, Jonghyon J, Cholsu R. An improved multi-directional interpolation for spatial error concealment. Multimedia Tools Appl 2019;78(2):2587–98.
[16] Khan S, Lee D-H, Khan MA, Siddiqui MF, Zafar RF, Memon KH, Mujtaba G. Image Interpolation via Gradient Correlation-Based Edge Direction Estimation. Scientific Programming 2020;2020.
[17] Song Q, Xiong R, Liu D, Xiong Z, Wu F, Gao W. Fast image super-resolution via local adaptive gradient field sharpening transform. IEEE Trans Image Process. 2018;27(4):1966–80.
[18] Zhu S, Zeng B, Zeng L, Gabbouj M. Image interpolation based on non-local geometric similarities and directional gradients. IEEE Trans Multimedia 2016;18(9):1707–19.
[19] Zhang K, Tao D, Gao X, Li X, Xiong Z. Learning multiple linear mappings for efficient single image super-resolution. IEEE Trans Image Process 2015;24 (3):846–61.

[20] Nayak R, Balabantaray BK, Patra D. A New Single-Image Super-Resolution Using Efficient Feature Fusion and Patch Similarity in Non-Euclidean Space. Arabian J Sci Eng 2020;45(12):10261–85.
[21] Fang Y., Shi M., Li X., Liu Y. Single image super-resolution based on gradient profile prior and nonlocal self-similarity feature, in: Eleventh International Conference on Digital Image Processing (ICDIP 2019), Vol. 11179, International Society for Optics and Photonics, 2019, p. 111794P.
[22] Xu Y, Li J, Song H, Du L. Single-Image Super-Resolution Using Panchromatic Gradient Prior and Variational Model. Math Problems Eng 2021;2021.
[23] Khan S, Lee D-H, Khan MA, Gilal AR, Mujtaba G. Efficient edge-based image interpolation method using neighboring slope information. IEEE Access 2019;7:133539–48.
[24] Yao X, Wu Q, Zhang P, Bao F. Adaptive rational fractal interpolation function for image super-resolution via local fractal analysis. Image Vis Comput 2019;82:39–49.
[25] Liu X, Zhao D, Zhou J, Gao W, Sun H. Image interpolation via graph-based Bayesian label propagation. IEEE Trans Image Process 2013;23(3):1084–96.
[26] Dougherty ER. Digital image processing methods. CRC Press; 2020.
[27] Kumar DP. Particle Swarm Optimization: The Foundation. In: Applying Particle Swarm Optimization. Springer; 2021. p. 97–110.
[28] Ali W, Khan MS, Hasan M, Khan ME, Qyyum MA, Qamar MO, Lee M. Introduction to Cuckoo Search and Its Paradigms: A Bibliographic Survey and Recommendations. AI and Machine Learning Paradigms for Health Monitoring System: Intelligent Data Analytics 2021;86:79.
[29] Soumaya Z, Taoufiq BD, Benayad N, Yunus K, Abdelkrim A. The detection of Parkinson disease using the genetic algorithm and SVM classifier. Appl Acoust 2021;171:107528.
[30] Singh H, Kommuri SVR, Kumar A, Bajaj V. A new technique for guided filter based image denoising using modified cuckoo search optimization. Expert Syst Appl 2021;176:114884.
[31] Parsopoulos MRKE, Martí P, Panos P. Particle Swarm Methods. 2018.
[32] Kundu R, Chakrabarti A, Lenka P. A Novel Technique for Image Denoising using Non-local Means and Genetic Algorithm. National Academy Science Letters 2021:1–7.
[33] Nair RR, Singh T. MAMIF: multimodal adaptive medical image fusion based on B-spline registration and non-subsampled shearlet transform. Multimedia Tools Appl 2021:1–27.
[34] Bizzarri M, Lávička M. Interpolation of Hermite data by clamped Minkowski Pythagorean hodograph B-spline curves. J Comput Appl Math 2021;392:113469.
[35] Zhang W, Zhao Y. Hierarchical Registration of Brain Images Based on B-splines and Laplacian Commutators. Optik 2021:167022.
[36] Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 2004;13(4):600–12.