

# Stochastic Rounding for Image Interpolation and Scan Conversion

Olivier Rukundo

[orukundo@gmail.com](mailto:orukundo@gmail.com)

## ABSTRACT

The stochastic rounding (SR) function is introduced to demonstrate the effects of stochastically rounding row and column subscripts on image interpolation quality in nearest neighbor interpolation (NNI). The introduced SR function is based on a pseudorandom number that enables the pseudorandom rounding up or down of any non-integer row and column subscripts. Also, the SR function exceptionally enables rounding up of any possible cases of subscript inputs that are inferior to a pseudorandom number – especially at a high interpolation scaling ratio. The quality of NNI-SR interpolated images is evaluated against the quality of reference images – before and after applying smoothing and sharpening filters, mentioned. The quality of NNI-SR interpolated scan conversion video frames is evaluated without using any references – focusing on the quality of one frame after every 78-milliseconds for 10 000 milliseconds. Relevant experimental simulation results, discussions, and recommendations are also provided.

**Keywords:** cardiac ultrasound, deterministic rounding, image quality, interpolation, pseudorandom number, scan conversion, stochastic rounding, video quality.

## 1. INTRODUCTION

Digital image interpolation pervades many engineering applications where estimates of image values at points other than the original grid are required and/or affects the desired results and/or the way to obtain them [1]. There exist many image interpolation algorithms, in various categories, that were developed focusing on improving the accuracy or efficiency of the algorithm, depending on target applications including but not limited to rescaling, reslicing, rendering, zooming, coordinate transformations in two-dimensional data or scan conversion, tomographic reconstruction and image registration [1]. One of the most recent applications of interest is the artificial intelligence (AI)-based image super-resolution [2],[3]. For example, in [2], the authors' core idea was to enhance the quality of the bilinear interpolation images, by applying a set of pre-learned filters on the image patches, chosen by an efficient hashing mechanism.

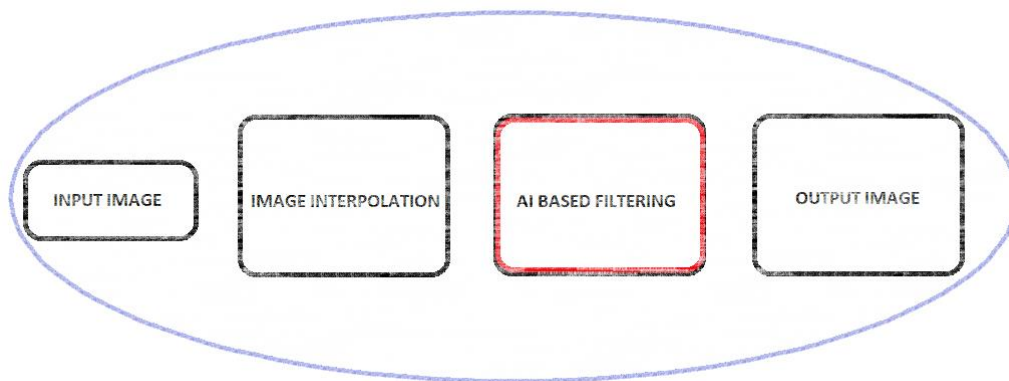


Figure 1: Generalizable steps for AI-based image super-resolution enhancement.

In another example, presented in [3], authors used the bicubic image interpolation algorithm to upscale the input or source image to meet the same size as the reference image before starting to recover from it, a resolution enhanced image comparable to the ground truth high-resolution image. In brief, generalizable steps followed by authors, in [2] and [3], can be illustrated as shown in Figure 1. Both bilinear and bicubic interpolation algorithms belong to the extra pixel category [4]. This category encompasses all image interpolation algorithms that create non-original pixels to achieve interpolation

results [4]. To the best of the author's knowledge, the effects of the image interpolation algorithms belonging to the non-extra pixel category are not yet evaluated in AI-based image super resolution. Today, the widely known interpolation algorithm falling in the non-extra pixel category is nearest neighbor image interpolation (NNI) [4]. NNI does not create non-original pixels to achieve interpolation results [5]. In addition, the NNI remains the fastest image interpolation algorithm [6], [7]. However, NNI has an inherent flaw of producing heavy jagged artefacts at object edges in some images, especially those with grain textures or similar. For the moment, the author does not aim at evaluating or using the NNI algorithm in one of the upscaling steps shown in Figure 1.

First, in this work, the author demonstrates the effects of stochastically rounding row and column subscripts on image interpolation quality in NNI. As a reminder - in the source image, the pixel coordinate subscripts are all integers. Therefore, selection of pixels from the source image may only be possible if subscripts in the destination image are all integers. NNI algorithm is based on the deterministic pixel selection, which is also based on a rounding function of interest. In [4], the author demonstrated that the best deterministic rounding (DR) function for nearest neighbor image interpolation purposes was the *ceil* function. However, when the non-integer's fractional part equals a half unity, the question about justification of the gain or loss of a half unity when the *ceil* function or any other DR function is used remains open. Therefore, in this work, the author also attempts to answer that question by also introducing the SR function in the NNI algorithm.

SR is an old idea that was proposed in the 1950s and 1960s by Forysthe and Hull and Swenson [8]. According to authors, in [8], this SR idea is attracting renewed interest in artificial intelligence/deep learning because it can improve the accuracy of the underlying computations. For  $x \in \mathbb{R}$  with  $x \notin F$  (where  $F \subseteq \mathbb{R}$  denotes the floating-point number system), authors considered 2 stochastic rounding modes shown in Eq.1 and Eq.2, [8]:

$$\textbf{Mode-1: } fl(x) = \begin{cases} \lceil x \rceil & \text{with probability } 0.5 \\ \lfloor x \rfloor & \text{with probability } 0.5 \end{cases} \quad (1)$$

$$\textbf{Mode-2: } fl(x) = \begin{cases} \lceil x \rceil & \text{with probability } p = (x - \lfloor x \rfloor) / (\lceil x \rceil - \lfloor x \rfloor) \\ \lfloor x \rfloor & \text{with probability } 1 - p \end{cases} \quad (2)$$

In the first mode or Eq.1, authors round  $x \in \mathbb{R}$  with  $x \notin F$  up or down with equal probability to the respective nearest floating-point number. In the second mode or Eq.2, authors round with a probability that is 1 minus the relative distance of  $x$  to each of the nearest floating-point numbers. For  $x \in \mathbb{R}$ ,  $\lfloor x \rfloor = \max\{y \in F: y \leq x\}$ ,  $\lceil x \rceil = \min\{y \in F: y \geq x\}$ , so that  $\lfloor x \rfloor \leq x \leq \lceil x \rceil$  with equality throughout if  $x \in F$ . For  $x \notin F$ ,  $\lfloor x \rfloor$  and  $\lceil x \rceil$  are adjacent floating-point numbers. More details are provided in [8].

Secondly, the author extends the application of the NNI-SR algorithm to interpolated scan conversion operations to evaluate the effects on video frames. Scan conversion is an operation that aims at translating input data, captured in different coordinates, into Cartesian coordinates, which are more suitable for display [9]. More information on the scan conversion system block diagram and the example of scan conversion using bilinear interpolation for sectorized images are provided in [9], [10].

The rest of the paper is organized as follows: Part two introduces the SR function. Part three presents numerical examples showing the comparison of results of SR and DR functions. Part four presents experimental results. Relevant discussions are provided in Part five. The conclusion is given in Part six.

## 2. STOCHASTIC ROUNDING FUNCTION

The stochastic rounding function is developed based on the equation that incorporates a *rand function* found in MATLAB. The MATLAB *rand* function is based on a new pseudorandom number generator named Mersenne Twister (MT). According to [11], the MT pseudorandom number generator seems to be the best among all generators ever implemented, with the period  $2^{19937} - 1$  and 623-dimensional equidistributional property. Also, the success of this C-Code MT19937 has been achieved thanks to two new ideas added to the previous version, Generalized Feedback Shift Register (GFSR), namely (1) the incomplete array, and (2) the inversive-decimation method [11]. In the experimental simulations, presented in this work, the pseudorandom number (*r*) was rounded to one digit. Also, the pseudorandom number was tuned to randomly vary between 0 and 0.5. In this way, it was possible to automate probabilities of stochastically rounding up or down thus achieving non-zero positive integers to be used as row and column subscripts of pixel coordinates. Note that, due to the intended application - of rounding non-integer row and column subscripts - Eq.3 incorporates conditions that allow it to only output non-zero positive integers.

$$sr(x) = \begin{cases} [x - r], & \text{if } x > r \wedge (x \bmod 1) > 0 \\ [x], & \text{otherwise} \end{cases} \quad (3)$$

In this way, the first condition ensures that any input index or subscript is greater than any pseudorandom number varying between 0 and 0.5. The second condition ensures that any input subscript is of non-integer type before proceeding to randomly rounding up or down. Note that, for  $0 \leq r \leq 0.5$ , it is an exception if  $r > x$ . Therefore, in such exceptional case, the SR function rounds up (e.g., see Table 1: See the first line in the 4X group).

The following Figure 2 shows an example of destination pixel coordinates subscripts before *round-off* operations. In Figure 2, (a) and (b) results were obtained by doubling a 3-by-3 matrix and plotting the coordinates of the matrix elements. Note that, when the rounding operation is random in NNI - this means a stochastic pixel selection.

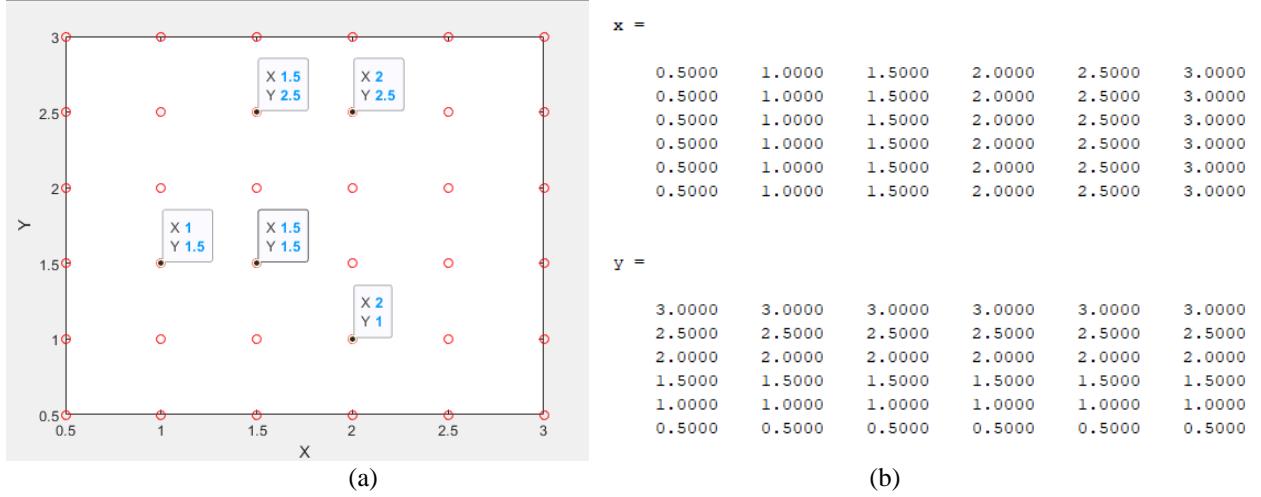


Figure 2: (a) pixel coordinate subscripts before round-off operations. (b) row (y) and column (x) subscripts before round-off operations.

In Table 1, the author compares the SR and DR functions after upscaling row/column subscripts of coordinates of the elements two times, three times, and four times with reference to the original size of the 3-by-3 matrix. As can be seen, Table 1 shows a column of subscripts, a column of random numbers, a column of DR results, and a column of SR results as well as columns of the elapsed time in both cases. In Table 1, it can be seen that when the scaling ratio is equal to two, the SR results differ from the DR results, twice. The same happens when the scaling ratio is equal to three. When the scaling ratio is equal to four, the SR results differ from the DR results, three times, except that in this case, there is an exception, mentioned earlier - about when  $x < r$ . Note that, SR may also produce results similar to DR results, but that is not guaranteed because the SR's output relies on the pseudorandom value. Also, it is important to note that, the results presented in Table 1 are specific to a particular case of  $r$  value and input numbers.

The pseudo-randomness of the SR function answers the question related to how to round a non-integer subscript in NNI (or in any other application area) when the fractional part is equal to 0.5. As can be seen, in the first case involving 2X (1.5 and 2.5), the SR behaved like the floor function, instead of the traditional otherwise. In the second case involving 4X (1.5, 2.5, and 3.5), the SR behaved like the ceil function. Comparing the elapsed time or line reading time, it can be seen that the time taken by both the SR and DR functions (to round a given series of non-integers) is too small to make any significant difference.

Note that, the most interesting point between DR and SR functions is that, normally when the fraction part of a non-integer equals 0.5, the DR function always rounds the non-integer to the next integer, which is not always the case with the SR function.

### 3. NUMERICAL EXAMPLES

Table 1:  $x$  represents row or column subscripts.  $r$  represents a random number rounded to one digit. Here,  $DR(x) = \text{ceil}(x)$ .

<i>ratio</i>	<i>x</i>	<i>r</i>	DR ( <i>x</i> )	SR ( <i>x</i> )	DR( <i>sec</i> )	SR( <i>sec</i> )
2X	0.5000	0.4	1	1	0.18 * 1.0e-05	0.68 * 1.0e-05
	1.0000	0.5	1	1		
	<b>1.5000</b>	0.5	<b>2</b>	<b>1</b>		
	2.0000	0.1	2	2		
	<b>2.5000</b>	0.5	<b>3</b>	<b>2</b>		
	3.0000	0.3	3	3		
3X	0.3333	0.1	1	1	0.15 * 1.0e-05	0.60 * 1.0e-05
	0.6667	0.4	1	1		
	1.0000	0.3	1	1		
	1.3333	0.5	<b>2</b>	<b>1</b>		
	1.6667	0.3	2	2		
	2.0000	0.2	2	2		
	2.3333	0.4	<b>3</b>	<b>2</b>		
	2.6667	0.3	3	3		
4X	3.0000	0.1	3	3	0.04 * 1.0e-05	0.17 * 1.0e-05
	<b>0.2500</b>	<b>0.5</b>	1	1		
	0.5000	0.1	1	1		
	0.7500	0.4	1	1		
	1.0000	0	1	1		
	1.2500	0.4	<b>2</b>	<b>1</b>		
	<b>1.5000</b>	0.4	2	2		
	1.7500	0.4	2	2		
	2.0000	0.4	2	2		
	2.2500	0.5	<b>3</b>	<b>2</b>		
	<b>2.5000</b>	0.1	3	3		
	2.7500	0.3	3	3		
	3.0000	0.4	3	3		
	3.2500	0.4	<b>4</b>	<b>3</b>		
	<b>3.5000</b>	0.4	4	4		
	3.7500	0.5	4	4		
	4.0000	0.1	4	4		

### 4. EXPERIMENTS

#### 4.1 Datasets, smoothing/sharpening method, IQA metrics

**a) Dataset:** Here, the used image dataset originated from the USC-SIPI Database of 210 Textures, Aerials, Miscellaneous, and Sequences images [12]. Here the author uses input images of  $128 \times 128$  size and reference images of  $512 \times 512$  size, all converted to 8bits using R2020a MATLAB. All experimental images are also available at the author's GitHub via [GitHub.com/orukundo](https://github.com/orukundo) [13].

**b) Smoothing/sharpening method:** The 2-D Gaussian smoothing kernel and sharpened using the unsharp masking methods - available in the MATLAB 2020a image processing toolbox - are used to extend or augment experiments via evaluating smoothed and sharpened interpolation results.

**c) IQA metrics:** In the beginning, only full-reference (FR) IQA metrics are used. Those included the mean-squared error (MSE), structural similarity index (SSIM), and peak signal to noise ratio (PSNR). These FR-IQA metrics are selected to quantify or measure the closeness or similarity of modified or distorted images (i.e., in this case, interpolated images) against their corresponding pristine images (i.e., reference images), [14]. Note that for SSIM and PSNR, normally when the scores are higher (closer to 1 and 100) that means the better visual quality. For MSE when the scores are lower (closer

to 0), that normally means better visual quality. Here, it is important to note that MATLAB's tic and toc command function is also used to check the elapsed time while reading code lines of the SR and DR functions (as shown in Table 1). In the end, - given that there exist no reference images or videos for cardiac ultrasound images or videos - the video frames quality assessment is done using no-reference (NR) IQA metrics. The selected NR-IQA metric of interest is the Perception-based Image Quality Evaluator (PIQE) [15], [16]. Specifically, PIQE is used to calculate the no-reference perceptual image quality of one frame after every 78-milliseconds for 10 000 milliseconds (i.e., entire video duration). This 78 milliseconds timestamp is estimated based on the number of frames of each video and the entire video duration as well as the suitability for graphical representation. To understand the PIQE scores, the quality scale and score range are as follows: Excellent [0  $\leftrightarrow$  20]. Good [21  $\leftrightarrow$  35]. Fair [36  $\leftrightarrow$  50]. Poor [51  $\leftrightarrow$  80]. Bad [81  $\leftrightarrow$  100], [16]. For scan conversion operations, T5D data files are used after being acquired from Duke University's Experimental Ultrasound System, T5, [10], [17]. More information on Duke University's Experimental Ultrasound System, T5 can be found via [18],[19],[20]. Note that, with T5D files, scan conversion operations are doable using a dedicated graphical user interface, developed in MATLAB, for post-processing of those ultrasound image sequences (from Duke University's Experimental Ultrasound System, T5).

#### 4.2 Evaluation of full image interpolation quality using full-reference IQA metrics

Table 2: Score estimates obtained using SSIM, PSNR, and MSE metrics during the evaluation of the NNI-SR and NNI-DR algorithms images against reference images of interest and before applying smoothing and sharpening filters

	SSIM		PSNR		MSE	
	NNI-DR	NNI-SR	NNI-DR	NNI-SR	NNI-DR	NNI-SR
IMAGE1	0.4809	0.5127	21.387	22.058	472.42	404.82
IMAGE2	0.5306	0.5688	21.443	22.391	466.41	374.34
IMAGE3	0.8713	0.8825	30.174	31.395	62.469	47.152
IMAGE4	0.8230	0.8388	23.667	24.903	279.46	210.24
IMAGE5	0.5555	0.5859	20.186	21.162	622.88	497.50
IMAGE6	0.5067	0.5480	20.606	21.586	565.45	451.30

Table 3: Score estimates obtained using SSIM, PSNR, and MSE metrics during the evaluation of the NNI-SR and NNI-DR algorithms images against reference images of interest and before applying smoothing and sharpening filters

	SSIM		PSNR		MSE	
	NNI-DR	NNI-SR	NNI-DR	NNI-SR	NNI-DR	NNI-SR
IMAGE1	0.5112	0.5481	21.381	22.334	473.03	379.89
IMAGE2	0.5627	0.6081	21.338	22.658	477.77	352.58
IMAGE3	0.8857	0.9050	29.910	31.727	66.38	43.68
IMAGE4	0.8358	0.8637	23.221	25.070	309.67	202.32
IMAGE5	0.5876	0.6354	20.009	21.528	648.86	457.30
IMAGE6	0.5333	0.5895	20.516	21.977	577.32	412.45

#### 4.3 Evaluation of cropped images interpolation quality against reference images of the same size

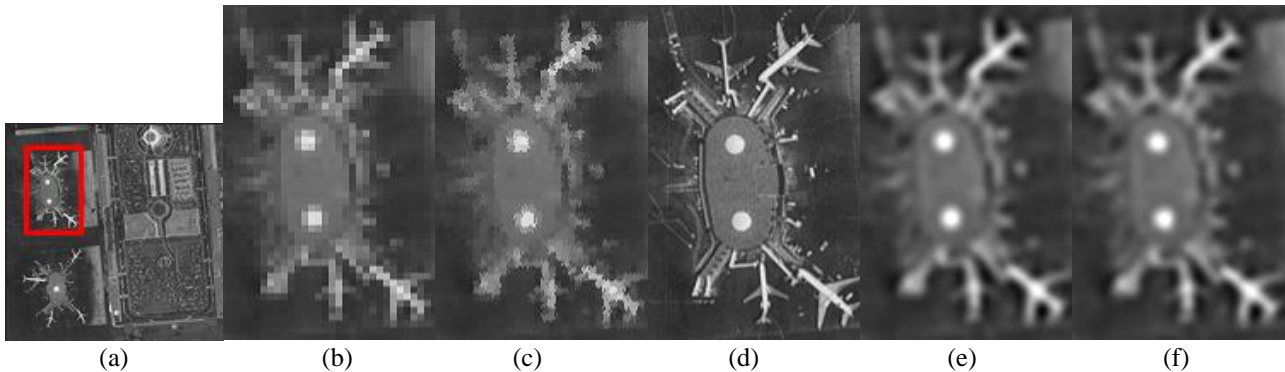


Figure 3: (a) Input *image1*. (b) cropped *image1* after NNI-DR interpolation. (c) cropped *image1* after NNI-SR interpolation. (d) cropped *image1* as the RF *image1*. (e) shows the (b)-filtered. (f) shows the (c)-filtered.

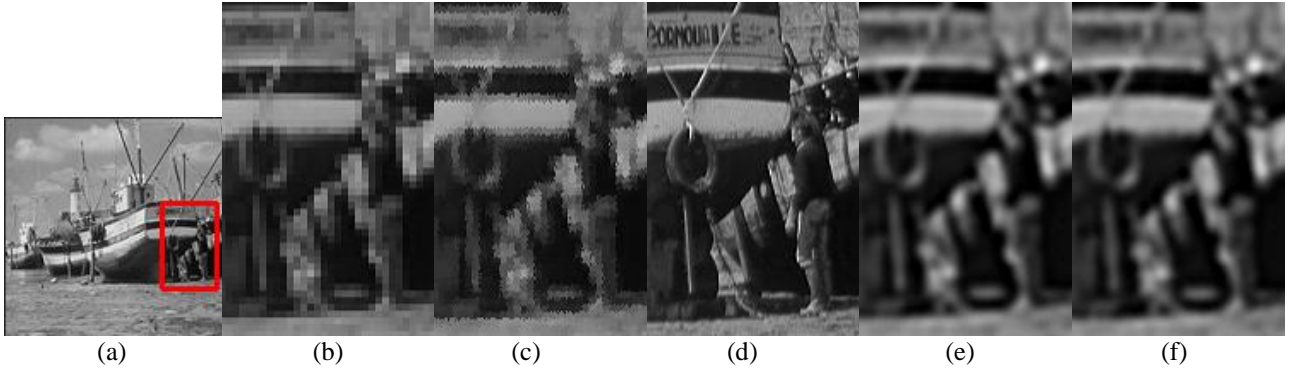


Figure 4: (a) Input *image2*. (b) cropped *image2* after NNI-DR interpolation. (c) cropped *image2* after NNI-SR interpolation. (d) cropped *image2* as the RF *image2*. (e) shows the (b)-filtered. (f) shows the (c)-filtered.

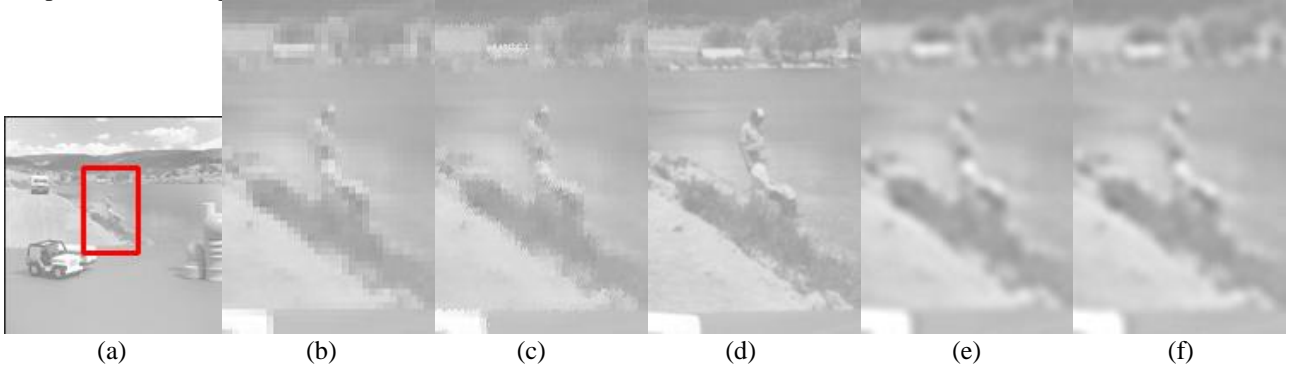


Figure 5: (a) Input *image3*. (b) cropped *image3* after NNI-DR interpolation. (c) cropped *image3* after NNI-SR interpolation. (d) cropped *image3* as the RF *image3*. (e) shows the (b)-filtered. (f) shows the (c)-filtered.

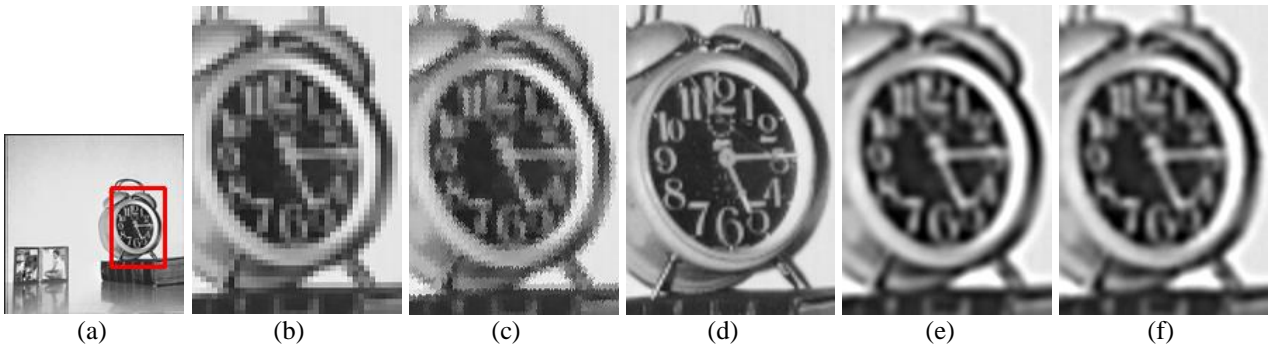


Figure 6: (a) Input *image4*. (b) cropped *image4* after NNI-DR interpolation. (c) cropped *image4* after NNI-SR interpolation. (d) cropped *image4* as the RF *image4*. (e) shows the (b)-filtered. (f) shows the (c)-filtered.

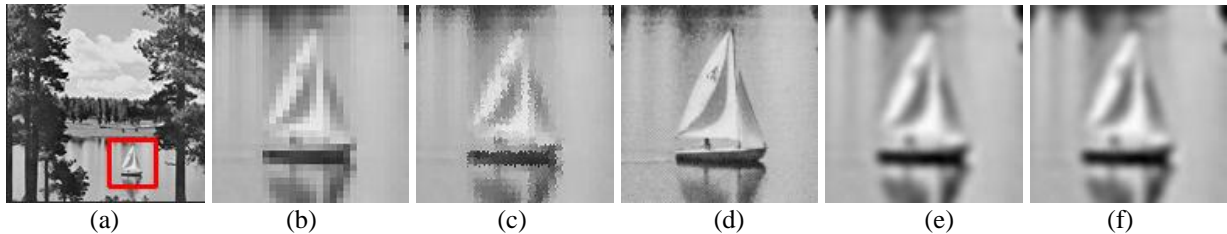


Figure 7: (a) Input *image5*. (b) cropped *image5* after NNI-DR interpolation. (c) cropped *image5* after NNI-SR interpolation. (d) cropped *image5* as the RF *image5*. (e) shows the (b)-filtered. (f) shows the (c)-filtered.





Figure 8: (a) Input *image6*. (b) cropped *image6* after NNI-DR interpolation. (c) cropped *image6* after NNI-SR interpolation. (d) cropped *image6* as the RF *image6*. (e) shows the (b)-filtered. (f) shows the (c)-filtered.

#### 4.4 Additional experiments to evaluate the quality of ultrasound video frames after interpolated scan conversion

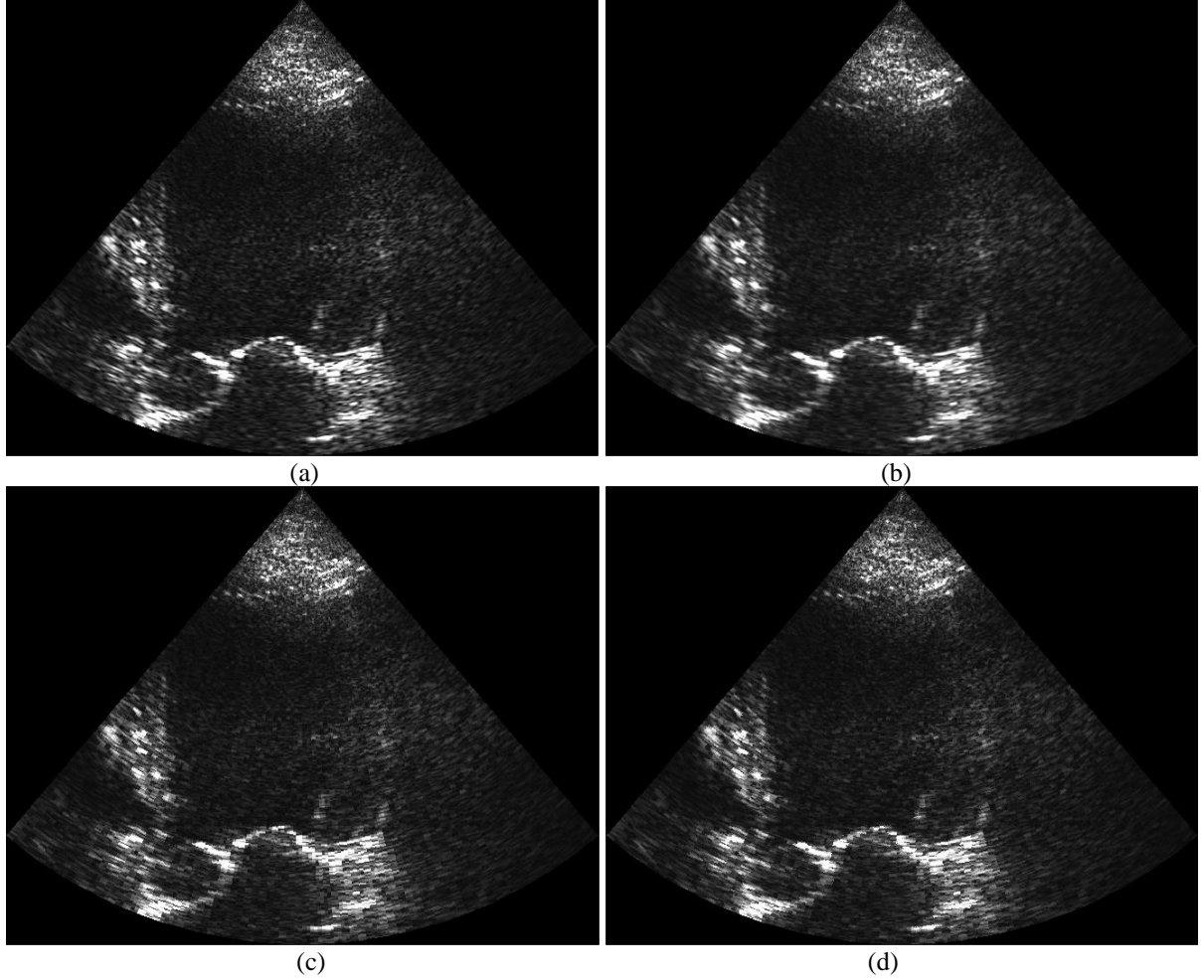


Figure 9: (a) bicubic, (b) bilinear, (c) NNI-DR, (d) NNI-SR (Also see < <https://github.com/orukundo/Interpolated-Scan-Conversion-of-B-Mode-Cardiac-Ultrasound-Image-Sequences> > for relevant videos: *Interpolated Scan Conversion - 60*)

## 5. DISCUSSIONS

Let us start by looking at the content of Table 3. Here, there are columns of score estimates achieved by NNI-DR and NNI-SR algorithms using the SSIM, PSNR, and MSE FR-IQA metrics. As can be seen, in all five images/cases, the NNI-SR algorithm achieved score estimates slightly higher than the NNI-DR score estimates relevant to SSIM and PSNR. Also, in all five images/cases, the NNI-SR algorithm achieved score estimates slightly lower than the NNI-DR score estimates

relevant to MSE. The same situation is repeated in Table 4, where, unlike in Table 3, the results presented are achieved after applying the smoothing and sharpening filters. The Table 3 comparison concludes that the NNI-SR performed better than the NNI-DR, in this specific situation. However, from the author's observation, it is also possible that the slight betterment of the NNI-SR may have been caused by the fact that the content of NNI-SR images was better aligned with the content of the reference images than the content of the NNI-DR images. Now, let us look at Figure 3: The (b) and (c) images have both produced jagged edges on the outline of the planes as well as the rest of the building. Also, comparing the results achieved by NNI-SR and NNI-DR against the (d) RF image, it is clear that only the NNI-SR algorithm reconstructed the white dots in a way almost similar to the way such dots look in the (d) image. Also, it is clear, the NNI-DR changed the three white dots or circles to squares.

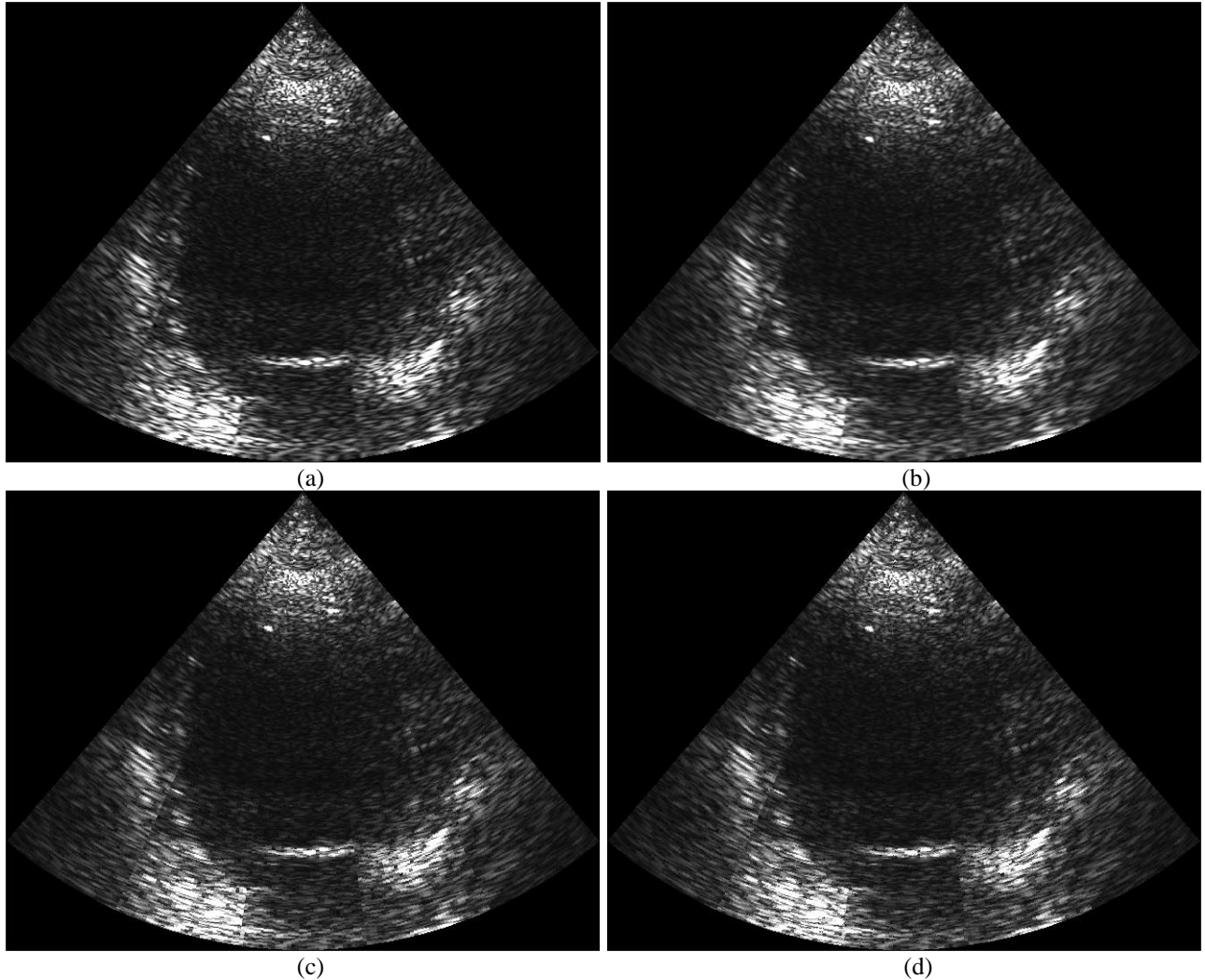


Figure 10: (a) bicubic, (b) bilinear, (c) NNI-DR, (d) NNI-SR (Also see < <https://github.com/orukundo/Interpolated-Scan-Conversion-of-B-Mode-Cardiac-Ultrasound-Image-Sequences> > for relevant videos: Interpolated Scan Conversion - 1000)

However, a similar situation did not repeat after smoothing and sharpening the results of NNI-DR and NNI-SR algorithms, shown in (e) and (f), respectively. Note that, in all these cases, getting a closer and clear view of the planes and/or their locations was not successful, even if the results suggested the existence of plans and airport terminal - even without having previously seen the RF image in (d). In Figure 4, the (b) and (c) images have shown jagged edges on the outline of the standing man as well as the rest of the ship edges. As can be seen, the NNI-DR algorithm produced so heavy jagged artefacts that the silhouette of the man disappeared completely, as shown in (b). Now, with the image produced by the



NNI-SR algorithm, in (c), the man's silhouette is only less hardly imaginable than in the NNI-DR case, shown in (b). After smoothing and sharpening, the results became too blurred that is impossible to imagine the man's silhouette, as shown in (e) and (f). If one sees the (e) and (f) images without having previously seen the RF image in (d), it is not possible to imagine the presence of a man's silhouette or ship edges. Here, it is also important to note that by cropping a small part of the image (a), the aim was to get a closer and clear view of the man. Considering Figure 5. Also, here, (b) and (c) images have both produced heavy jagged edges on the outline of the two men as well as the rest of the beach. This demonstrates that getting a closer and clear view of the two men was not successful using both NNI-DR and NNI-SR. But, after smoothing and sharpening the (b) and (c) images, the results shown in (e) and (f) allow one to imagine the presence of two men, one standing and one sitting, without even having previously seen the RF image in (d). A similar situation is repeated in Figure 6, Figure 7, and Figure 8. With Figure 9 and Figure 10, it can be seen that these two figures contain dissimilar sectorized images from the cardiac ultrasound imaging processes. In all cases, the bicubic interpolated scan conversion image (a) looks better than the rest of bilinear, NNI-DR, and NNI-SR interpolated scan conversion images in (b), (c), and (d), respectively. Here is important to note that originally the frame rate was 60 frame per second for images in Figure 9. In Figure 10, the frame rate was originally 1074 frame per second. All these frame rates could be visualized using the Graphical User Interface developed for that purpose. Note that relevant videos are available at <https://github.com/orukundo/Interpolated-Scan-Conversion-of-B-Mode-Cardiac-Ultrasound-Image-Sequences> (see the link entitled: Interpolated Scan Conversion-1000 and/or Interpolated Scan Conversion-60). Also, it is important to note that, to easily perceive the video quality difference – relevant to the mentioned interpolated scan conversion algorithms – the monitor resolution must be high enough. Further assessments of the results were achieved via plotting and comparing graphs of pixel intensity distributions in NNI-DR and NNI-SR interpolated images against the pixel intensity distribution in reference images – before and after filtering operations. As can be seen, Figure 11-(a) shows the number of pixels counts versus the corresponding number of bins in NNI-DR interpolated IMAGE1, NNI-SR interpolated IMAGE1 and RF IMAGE1.

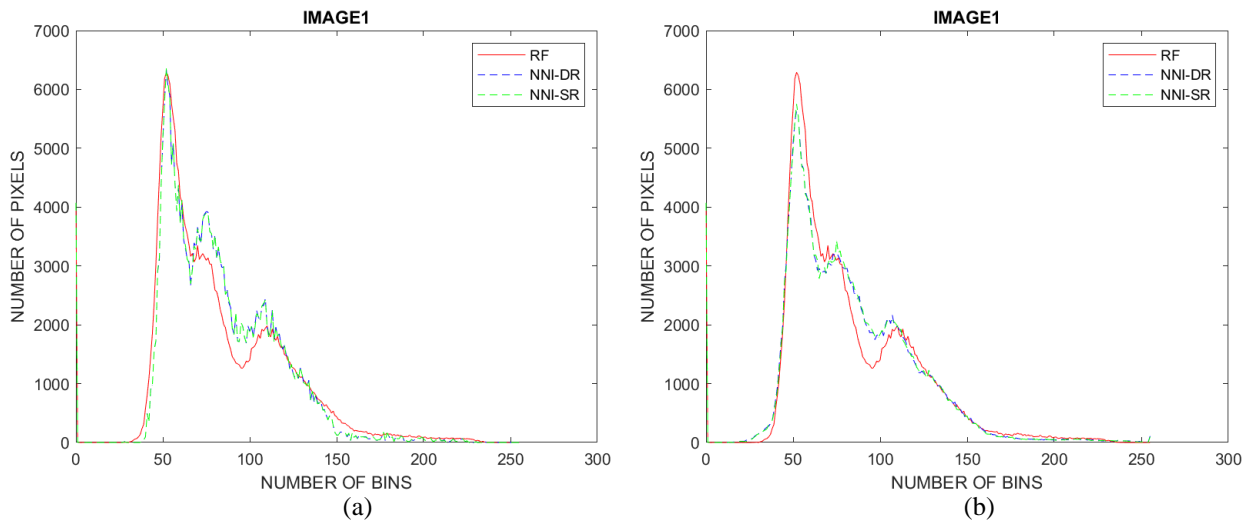


Figure 11: (a) before filtering. (b) after filtering

Figure 11-(b) shows the number of pixels counts versus the corresponding number of bins in NNI-DR interpolated and filtered IMAGE1, NNI-SR interpolated and filtered IMAGE1, and RF IMAGE1. As can be seen, in each case, none of the NNI-DR or NNI-SR results (represented by the blue and green line) matched perfectly with the RF results (represented by a red line). In other words, the number of pixels belonging to each bin of the RF IMAGE1 remained different from the number of pixels belonging to each bin of the NNI-DR and NNI-SR IMAGE1. This difference is also visible between NNI-DR and NNI-SR results, otherwise, it would not be possible to see the blue and green lines. Although not exactly at the same extent, a similar situation is generally repeated in Figure 12, Figure 13, Figure 14, Figure 15, and Figure 16 based on IMAGE2, IMAGE3, IMAGE4, IMAGE5, and IMAGE6. Note that the number of empty bins in the source image remained equal to the number of empty bins in the images interpolated by NNI-DR and NNI-SR (i.e., a condition in the non-extra pixel category). In the end, let us concentrate on Figure 17. In both (a) and (b) cases, the video frames were repeatedly evaluated after 78 milliseconds (instead of evaluating every frame), in each of the 10-seconds videos. This allowed us to get the idea of the performance of each scan conversion method mentioned. Later, this can allow us to

understand why at a specific time in a given video, the image or frame quality was bad or good in ultrasound systems. In the past, many attempts were done to develop methods to assess the quality of ultrasound imaging systems [21],[22],[23]. In the recent past, the author introduced an index for image interpolation quality assessment as a preliminary step to a suitable method for image quality assessment in ultrasound imaging – only focusing on undesirable artefacts, known as aliasing [24]. However, until now, the PIQE remains the only NR-IQA metric very sensitive to two interesting cases of undesirable artefacts in ultrasound imaging, namely: grain-like or speckle-like noise and blurriness. As can be seen, the bilinear interpolated scan conversion algorithm achieved the lowest mean score (i.e., blurriest video frames) in both (a) and (b) cases.

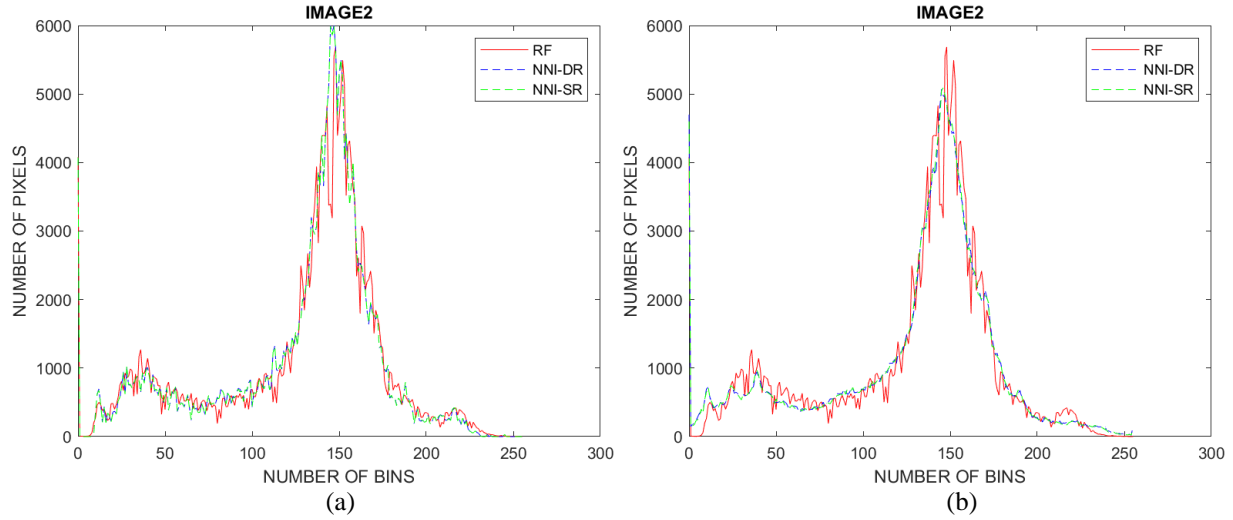


Figure 12: (a) before filtering. (b) after filtering

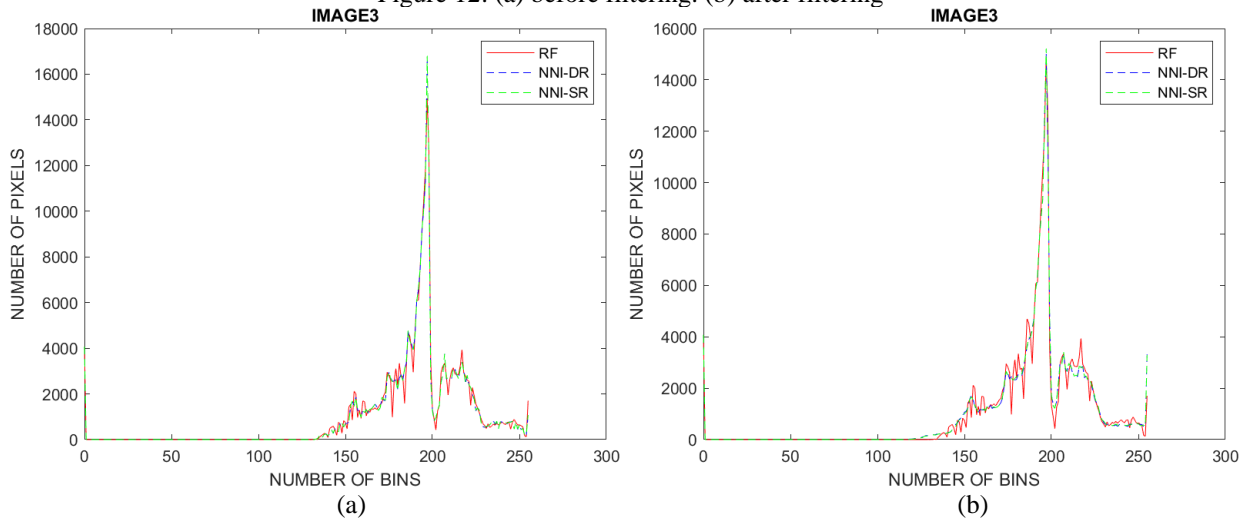


Figure 13: (a) before filtering. (b) after filtering

Note that, the current literature demonstrates that the bilinear interpolation has always been associated with being the most blurriness productive among all non-adaptive interpolation algorithms [25],[26],[27],[28] even if it has proved to be useful in other image processing techniques [29], [30], [31]. The fact that bilinear performed poorer than others in both (a) and (b) cases, confirms its inherent flaw of being the most interpolation blurriness productive. Now, separately Considering (b), where the original video frame rate was 60, both NNI-DR and NNI-SR interpolated scan conversion algorithms achieved the best PIQE scores compared to the other two interpolation algorithms of the extra-pixel category.

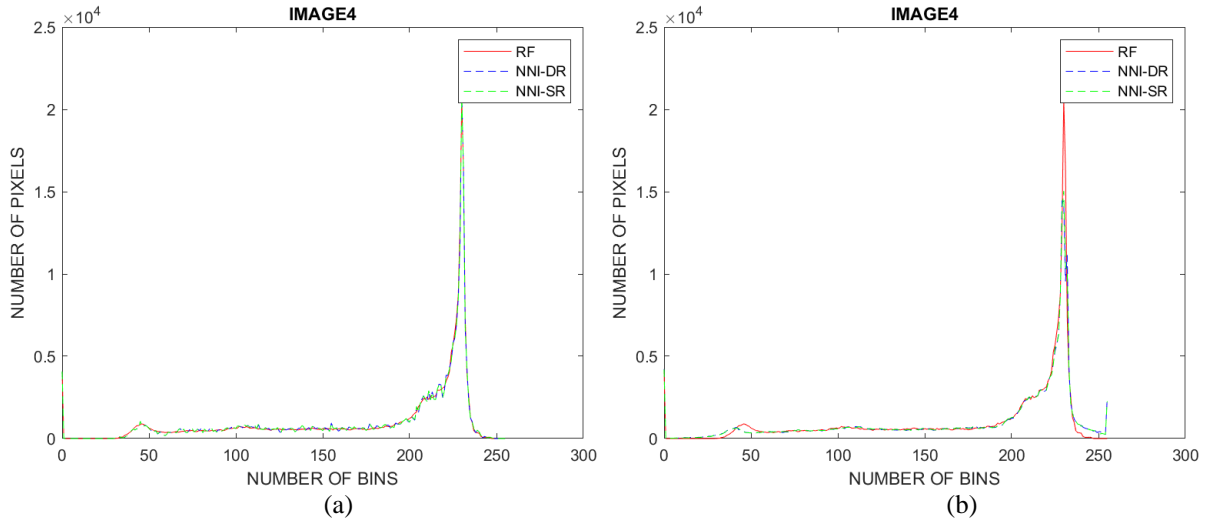


Figure 14: (a) before filtering. (b) after filtering

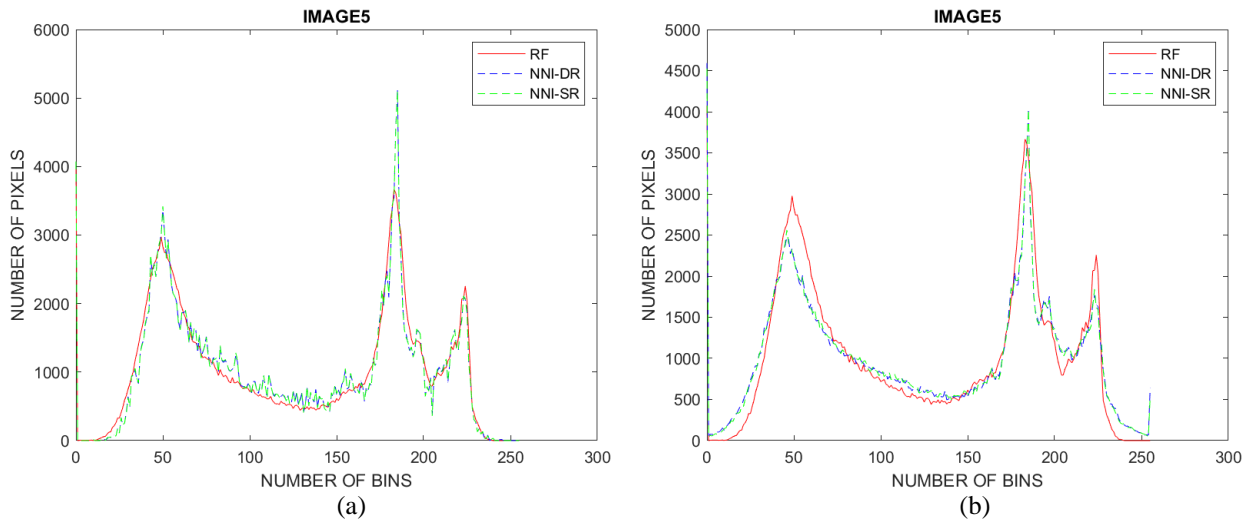


Figure 15: (a) before filtering. (b) after filtering

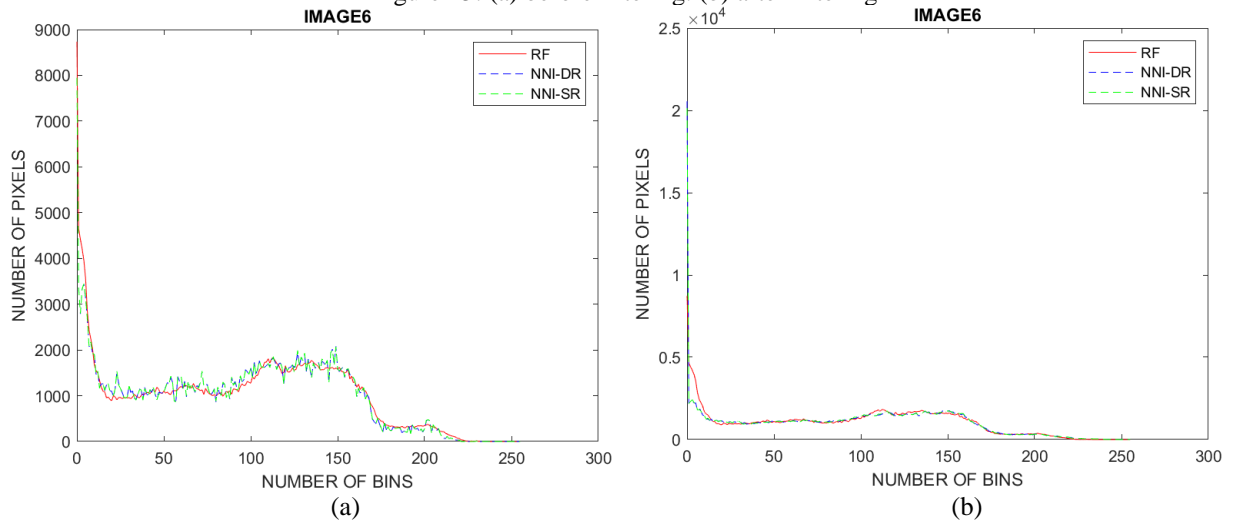


Figure 16: (a) before filtering. (b) after filtering

However, considering (a), where the original video frame rate was 1074, the bicubic interpolated scan conversion algorithm demonstrated strength that would normally be expected - as it normally produces better image quality than most non-adaptive interpolation algorithms [32],[33],[34],[35]. The mean scores were provided, in the legend, to quickly allow us to get the idea of the estimated performance of each of interpolated scan conversion algorithms. Note that, these mean scores are specific to these cases. Also, note that these mean score values may change, therefore they are not conclusive in these examples. It is important to also note that, the 78 milliseconds timestamp was adopted with reference to the frame rate and video duration to enable the more informative and clearer plotting of graphs – otherwise with only 10 seconds videos, the graphs would have looked like straight lines or similar. Note that the results in (a) and (b) also prove that the PIQE can be considered as the most suitable NR-IQA metric for ultrasound image quality assessment because, in (a) and (b) example, PIQE-based results match perfectly with the well-known performances of interpolation methods, mentioned.

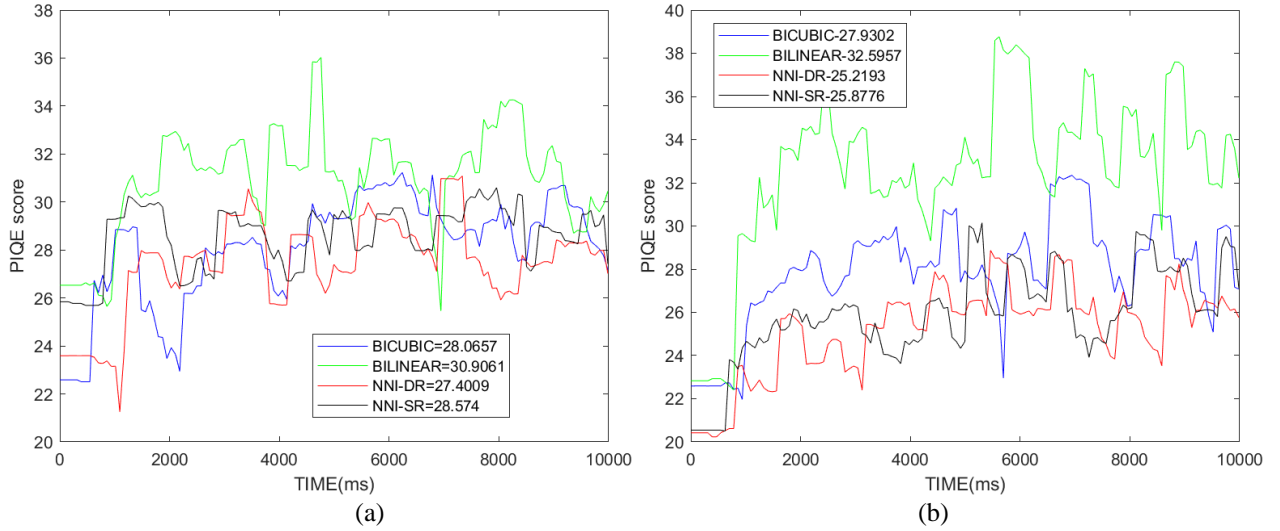


Figure 17: (a): Video1 - original frame rate equals 1074. (b) Video2 -original frame rate equals 60.

## 6. CONCLUSION

In this work, the author introduced the SR function in NNI to demonstrate the effects of stochastically rounding row and column subscripts on image interpolation quality. Furthermore, the author applied the resulting NNI-SR algorithm in interpolated scan conversion to evaluate the effects on the quality of video frames in cardiac ultrasound imaging. With the introduction of the SR function in NNI, the author tried to answer the question – raised by the traditional deterministic rounding way – about the reasonable rounding of the fractional part of the row or column subscript when it equals a half unity. Experimental results and discussions demonstrated close or comparable IQA metrics-based performances between the NNI-SR and NNI-DR algorithm - although differences were noticed in terms of edge artefacts produced by each algorithm. Experiments also demonstrated the possibility to apply the NNI-SR algorithm or SR function in other engineering areas – for example, in a data augmentation to create more deep learning training samples, etc.

## COMPETING INTERESTS

The author declares that he has no competing interests

## ACKNOWLEDGEMENT

Author would like to thank the reviewers and editors for constructive comments. Author would like to thank Duke University, Biomedical Engineering Department for support.

## REFERENCES

- [1] Handbook of Medical Image Processing and Analysis, 2nd Edition, I.N. Bankman (Ed.), Academic Press, ISBN 0123739047, San Diego, CA, USA, p. 984, (2008)
- [2] Romano, Y., et al., RAISR: Rapid and Accurate Image Super Resolution, In: ArXiv:1606.01299v3, (2016)
- [3] Dong, C., Loy, C.C., He, K., Tang, X., Learning a Deep Convolutional Network for Image Super-Resolution, In: Fleet D., Pajdla T., Schiele B., Tuytelaars T. (eds) Computer Vision, ECCV, (2014)
- [4] Rukundo, O., Non-extra Pixel Interpolation, International Journal of Image and Graphics, 20(4), 2050031, (2020)
- [5] Rukundo, O., Evaluation of Rounding Functions in Nearest Neighbour Interpolation, International Journal of Computational Methods, 10 pages, (2021)
- [6] Rukundo, O., Cao, H.Q., Nearest Neighbour Value Interpolation, International Journal of Advanced Computer Science and Applications, 3(4), (2012)
- [7] Rukundo, O., Maharaj, B.T., Optimization of Image Interpolation based on Nearest Neighbour Algorithm, International Conference on Computer Vision Theory and Applications, Lisbon, 641–647, (2014)
- [8] Connolly, M., Higham, N., Mary, T., Stochastic Rounding and its Probabilistic Backward Error Analysis, SIAM Journal on Scientific Computing, Society for Industrial and Applied Mathematics, 43 (1), A566–A585, (2021)
- [9] Li, X.H., Ultrasound Scan Conversion on TI's C64x+ DSPs, TI Application Report, SPRAB32, (2009)
- [10] Rukundo, O., Schmidt, S.E., Von Ramm, O.T., Software Implementation of Optimized Bicubic Interpolated Scan Conversion in Echocardiography, In: ArXiv: 2005.11269, (2020)
- [11] Matsumoto, M. Nishimura, T., Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator, ACM Transactions on Modeling and Computer Simulation, 8(1), 3–30, (1998)
- [12] USC-SIPI Image Database: <<http://sipi.usc.edu/database/database.php>>, Accessed: 2021-10-12
- [13] Modified-USC-SIPI-Image-Database: <<https://github.com/orukundo/Modified-USC-SIPI-Image-Database>>, Accessed: 2021-10-12
- [14] Rukundo, O., Schmidt, S., Extrapolation for Image Interpolation, Proc. SPIE 10817, Optoelectronic Imaging and Multimedia Technology V, 108171F, (2018)
- [15] Venkatanath, N., Praneeth, D., et al., Blind Image Quality Evaluation Using Perception Based Features, In Proceedings of the 21st National Conference on Communications (NCC). Piscataway, NJ: IEEE, (2015)
- [16] Sheikh, H. R., Wang, Z., et al., LIVE Image Quality Assessment Database Release 2, <<https://live.ece.utexas.edu/research/quality>>, Accessed: 2021-10-12
- [17] Moore, C., Castellucci, J., et al., Live High-Frame-Rate Echocardiography, IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, 62(10), 1779-1787, (2015)
- [18] Andersen, M.V, Moore, C., et al., Quantitative Parameters of High-Frame-Rate Strain in Patients with Echocardiographically Normal Function, Ultrasound Med Biol., 45(5), 1197-1207, (2019)
- [19] Andersen, M.V, et al., Feature tracking algorithm for circumferential strain using high frame rate echocardiography, 2016 Computing in Cardiology Conference (CinC), pp. 885-888, (2016)
- [20] Andersen, M.V, Moore, C., et al., High-Frame-Rate Deformation Imaging in Two Dimensions Using Continuous Speckle-Feature Tracking, Ultrasound Med Biol., 42(11), 2606-2615, (2016)
- [21] Hemmsen, M.C., Petersen, M.M, et al., Ultrasound image quality assessment: a framework for evaluation of clinical image quality, Proc. SPIE 7629, Ultrasonic Imaging, Tomography, and Therapy, 76290C, (2010)
- [22] Zhang, S., Wang, Y., et al., CNN-Based Medical Ultrasound Image Quality Assessment, Complexity, vol. 2021, Article ID 9938367, 9 pages, (2021)
- [23] Sassaroli, E., Crane, C., et al., Image quality evaluation of ultrasound imaging systems: advanced B-modes, J Appl Clin Med Phys., 20(3), 115–124, (2019)
- [24] Rukundo, O., Schmidt, S., Aliasing Artefact Index for Image Interpolation Quality Assessment, Proc. SPIE 10817, Optoelectronic Imaging and Multimedia Technology V, 108171E, (2018)
- [25] Rukundo, O., Effects of Improved-Floor Function on the Accuracy of Bilinear Interpolation Algorithm, Computer and Information Science, 8(4), (2015)
- [26] Rukundo, O., Schmidt, S., Effects of Rescaling Bilinear Interpolant on Image Interpolation Quality, Proc. SPIE 10817, Optoelectronic Imaging and Multimedia Technology V, 1081715, (2018)
- [27] Rukundo, O., Wu, K.N., and Cao, H.Q., Image Interpolation based on the Pixel Value corresponding to the Smallest Absolute Difference, IEEE Int. Workshop on Advanced Computational Intelligence, 432–435, (2011)
- [28] Rukundo, O., Normalized Weighting Schemes for Image Interpolation Algorithms, In: ArXiv: 2011.08559, (2020)



- [29] Rukundo, O., Pedersen, M., Hovde, Ø., Advanced Image Enhancement Method for Distant Vessels and Structures in Capsule Endoscopy, Computational and Mathematical Methods in Medicine, (2017)
- [30] Rukundo, O., Half-Unit Weighted Bilinear Algorithm for Image Contrast Enhancement in Capsule Endoscopy, International Conference on Graphic and Image Processing, 106152Q-1 - 106152Q-9, SPIE, (2018)
- [31] Rukundo, O., Effects of Empty Bins on Image Upscaling in Capsule Endoscopy, International Conference on Digital Image Processing, 104202P-1 - 104202P-8, SPIE, (2017)
- [32] Hwang, J.W., Lee, H.S., Adaptive image interpolation based on local gradient features, in IEEE Signal Processing Letters, 11(3), 359-362, (2004)
- [33] Li, Y., Qi, F., Wan, Y., Improvements On Bicubic Image Interpolation, 4th Advanced Information Technology, Electronic and Automation Control Conference, 1316-1320, (2019)
- [34] Li, X., Orchard, M. T., New edge-directed interpolation, in IEEE Trans. on Image Proc., 10(10), 1521-1527, (2001)
- [35] Tam, W.S., Kok, C.H., Siu, W.C., Modified edge-directed interpolation for images, Journal of Electronic Imaging 19(1), 013011, (2010)