Trade-off between Image Quality and Computational Complexity: Image Resizing Perspective

Iwan Setiawan[1*], Rico Dahlan[2], Akbari Indra Basuki[1], Heru Susanto[1], and Didi Rosiyadi[1]

[1]*Research Center for Data Sciences and Information, National Research and Innovation Agency*

[2]*Research Center for Telecommunication, National Research and Innovation Agency*
Jl. Cisitu No.21/154D, Bandung, 40135, Indonesia

[*]*Corresponding author. Email: iwan022@brin.go.id*

Abstract— This study proposed a new approach for resizing image deal with quality and computational complexity. Here, previous methods in image resizing do analytical works to approximate the original picture element (pixel) or to remove high frequency coefficients. For images with huge pixel, this will result in computational burden due to number of multiplication and addition in the synthesized formula. Instead of the works, this study proposed a new approach in removing the coefficients by exploiting the second-order block matrix without the need to synthesize the formula. It can be called a fully numeric image resizing method. The result shows that the resized version of original image has peak signal to noise ratio (PSNR) equal to 35.24 dB for resizing the famous Lena image which means compareable to the conventional which has PSNR value around 35 dB but here deriving analytical formula is not required. Reducing computational complexity is also achieved as expected with result only 16 addition involved with no multiplication required. This is lower than the conventional in term of computational complexity. Overall, the proposed method has a good balance for both performances than the conventional approaches.

Keywords— computational complexity; filtering formula; image quality; image resizing; resizing matrix

I. Introduction

In many applications, original digital images are required to be resized (up/downsized) [1]. This means the new images will have a lower/higher rank of matrix. Here the process achieved, through synthesizing formula to approximate the resized picture element (pixel) or by removing the high frequency coefficient conventionally. Moreover, the first option, the approximation approach, requires analytical work which involves multiplication and addition and works on spatial domain. The simplest one could be averaging elements of the block matrix. This oldest conventional tool is called bilinear interpolation. While other efforts in approximating the resized pixel involve more complex works such as a new fashion of bilinear interpolation [2]-[4], exploiting Benford's law to determine DCT coefficient [5], determining scaling factor [6], and detecting the edge distortion [7].

Another option is by transforming the pixel to frequency domain continued by performing up/down-sampling method in this new domain to remove the high frequency coefficient [8]. Regarding this approach, the works are about exploiting DCT low frequency components and sparsely in high frequency components [9], remapping high frequencies to the represent able range of the down-sampled spectrum [10], and exploiting invertible bijective transformation to mitigate the ill-posed due to removing the high frequency [11]. Performance of both approaches are measured by image quality of the resized image and computational complexity in delivering the resized image. In order to get the best performance of both parameters, learning-based methods are available but this smart approach is hard to implement due to hardware cost [12] such as to provide high performance computing equipment.

Instead of upsizing, this research will focus on downsizing. The available works on image downscaling are for example decimating high resolution pixel covered by the re-sampling kernel [13], proposing luma aware chroma down-sampling to improve image quality [14], exploiting the depth map down-sampling and coding scheme to minimize the distortion [15], the method which consider effect of in-camera downsizing on camera ID verification [16], the method which maintain accuracy of object detection in the context of self-driving vehicles [17], introduce a novel L_0-regularized optimization framework for image downscaling [18], and perform decreasing sample rate to the block to be down-sampled [19]. The objective of this research is to obtain a better trade-off between image quality and computational complexity with no additional hardware cost.

Additionally, discussion about resizing the compressed image is out of scope of this paper. This means the original image is assuming available for this purpose. By using our approach, highly number of multiplication and addition found in the analytical work can be avoided which means this proposed method will have a lower computational complexity than the conventional. For measuring quality of the resized image, ratio between peak signals to noise (PSNR) will be used [20].

Received 12 July 2022, Accepted 3 October 2022, Published 10 October 2022.

DOI: https://doi.org/10.15294/jte.v14i1.37629

II. METHOD

The conventional algorithm such as approximation method and removing the high frequency coefficient requires analytical effort to synthesize the formula with objective to remove matrix element with high frequency coefficient. By removing the coefficient, downsized version of the original image will be obtained with a good computational complexity. However, the image quality will be degraded as explained earlier in introduction part. In this section, the following algorithm is introduced to do down/up-sampling by exploiting the second-order block matrix without the need to do the analytical work. While for measuring the performance, algorithm to measure PSNR value is included. Figure 1 shows how to obtain downsized version of the original image.

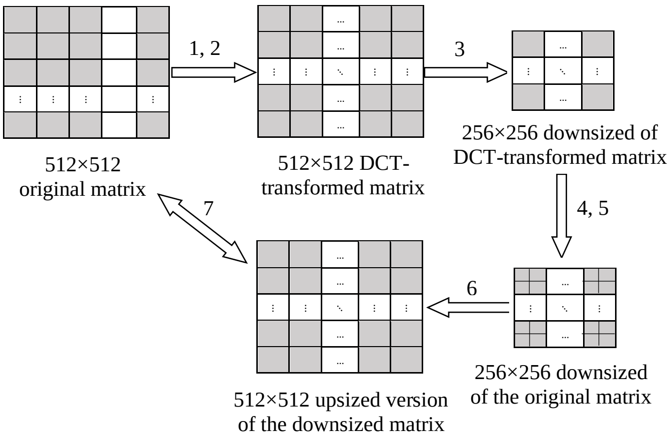


Figure 1. Diagram of the proposed method

The following algorithm is proposed:

1) Apply 2×2 block matrix to the original image

By exploiting the second-order block matrix, removing the high frequency coefficient is possible to do as soon as the transformation. How to implement the block to the original image is shown in Figure 2.

```
1. Read the n × n matrix element of the image.
   FOR first to n^th matrix element of original matrix
     set a blank cell with 2 × 2 block matrix
     FOR each block
         address 1 to 2 rows every time
         address 1 to 2 column every time
     END FOR
   END FOR
2. Get the original image with 2×2 block matrix
```

Figure 2. Pseudo-codes for applying 2×2 block matrix to the original image

2) Apply DCT to each 2×2 block matrix

The following algorithm shown in Figure 3 should be applied to compute DCT-transformed for the whole matrix, the original matrix should be divided to 2×2 block matrix, discussion about this DCT approach could be found in [21].

```
1. Read the original image with 2×2 block matrix
     FOR each block
         read element matrix
         compute DCT values
     END FOR
2. Get the DCT-transformed matrix
```

Figure 3. Pseudo-codes for applying 2×2 DCT matrix to the original image

3) Pick a pixel with low frequency

In order to get resized version of DCT-transformed matrix, the low frequency coefficient for each block is selected as the most significant bit. This can be achieved by performing algorithm as shown in Figure 4.

```
1. Select the low frequency coefficient for each 2×2 block
   matrix
     FOR each block
         read the coefficient
     END FOR
2. Get the resized version of DCT-transformed matrix
```

Figure 4. Pseudo-codes for the resized matrix

4) Apply zero padding to the inverse DCT

This step is required to return matrix element back to spatial domain. The resulted matrix will be obtained by performing algorithm as shown in Figure 5.

```
1. Read the resized image
     FOR each block
         read element matrix
         compute IDCT values
     END FOR
2. Read the inverse matrix.
   FOR first to n^th matrix element of concatenated matrix
     set a blank cell with 2 × 2 block matrix
   END FOR
```

Figure 5. Pseudo-codes for inverse DCT matrix

5) Get the down-sampled version of the original

To get down-sampled version of the original image, add zero value to each matrix element and compute the corresponding block matrix. By following the algorithm shown in Figure 6, down-sampled version of the original image will be obtained.

```
1. Read the inverse matrix with zero padding.
     FOR each block
         address 1 to 2 rows every time
         address 1 to 2 column every time
         compute DCT values
     END FOR
2. Get the down-sampled version of the original image
```

Figure 6. Pseudo-codes for the down-sampled image

6) Perform up-sampling to the down-sampled image

Before measuring PSNR value, downsized version of the original image is required to be upsized because both original and downsized images must have matrix with the same dimension to be compared. The following algorithm as shown in Figure 7 is intended to obtain the up-sampled version of the downsized image.

```
1. Read the down-sampled image.
   FOR first to n^th matrix element of down-sampled matrix
     set a blank cell with 2 × 2 block matrix
     FOR each block
         address 1 to 2 rows every time
         address 1 to 2 column every time
         add zeros values
     END FOR
   END FOR
2. Get the up-sampled version of the down-sampled image
```

Figure 7. Pseudo-codes for the up-sampled image

7) Measuring performance

In order to measure how good the downsized image, it is required to compute the ratio between peak signal to noise (PSNR) of the original image and up-sampled version of the resized image. The PSNR value depends on mean square error between the original and the up-sampled version of the resized image. The algorithm as shown in Figure 8 is intended to determine the PSNR.

```
1. Set original image as reference
2. Set up-sampled image as compared image
3. Compute PSNR
4. Get the PSNR values
```

Figure 8. Algorithm to measure PSNR

III. RESULTS AND DISCUSSION

Let an original image is provided and represented by the matrix shown in (1). If Lena as shown in Figure 9 is used as the original image, then (1) will have a new fashion as shown in (2).

$$\mathrm{I}_{\mathrm{image}} = \begin{bmatrix} \mathrm{I}_{(1,1)} & \mathrm{I}_{(1,2)} & \cdots & \mathrm{I}_{(1,n-1)} & \mathrm{I}_{(1,n)} \\ \mathrm{I}_{(2,1)} & \mathrm{I}_{(2,2)} & & \mathrm{I}_{(2,n-1)} & \mathrm{I}_{(2,n)} \\ \vdots & & \ddots & \vdots & \\ \mathrm{I}_{(n-1,1)} & \mathrm{I}_{(n-1,2)} & \cdots & \mathrm{I}_{(n-1,n-1)} & \mathrm{I}_{(n-1,n)} \\ \mathrm{I}_{(n,1)} & \mathrm{I}_{(n,2)} & & \mathrm{I}_{(n,n-1)} & \mathrm{I}_{(n,n)} \end{bmatrix}_{n\times n} \quad (1)$$

Figure 9. The original Lena image with size 512×512

$$I_{original} = \begin{bmatrix} 161 & 162 & \cdots & 155 & 127 \\ 162 & 163 & & 157 & 128 \\ \vdots & & \ddots & \vdots & \\ 43 & 47 & \cdots & 106 & 111 \\ 43 & 48 & & 102 & 110 \end{bmatrix}_{512\times 512} \quad (2)$$

The following results are obtained by performing the step-by-step algorithm as explained earlier in discussion about method.

1) Result for performing 2×2 block matrix to the original image

The result as shown in (3) will be obtained by performing 2 × 2 block to all matrix elements of the original (2) starting from the red block which contains matrix element of first-second row and first-second column, ended by the purple which has matrix elements of the last two-rows and columns.

$$I_{image} = \begin{bmatrix} \boxed{\begin{matrix} 161 & 162 \\ 162 & 163 \end{matrix}} & \cdots & \boxed{\begin{matrix} 155 & 127 \\ 157 & 128 \end{matrix}} \\ \vdots & \ddots & \vdots \\ \boxed{\begin{matrix} 43 & 47 \\ 43 & 48 \end{matrix}} & \cdots & \boxed{\begin{matrix} 106 & 111 \\ 102 & 110 \end{matrix}} \end{bmatrix}_{512\times 512} \quad (3)$$

2) Result for performing DCT to each 2×2 block matrix

To deliver the result as shown in (4), compute DCT values for each block as shown earlier in (3).

$$I_{DCT} = \begin{bmatrix} \mathbf{324} & -1 & \cdots & \mathbf{283.5} & 28.5 \\ -1 & 0 & & -1.5 & -0.5 \\ \vdots & & \ddots & \vdots & \\ \mathbf{90.5} & -4.5 & \cdots & \mathbf{214.5} & -6.5 \\ -0.5 & 0.5 & & 2.5 & 1.5 \end{bmatrix}_{512\times 512} \quad (4)$$

3) Result for picking a pixel with low frequency

Now, time to select the low frequency coefficient for each block. By performing this, the size of matrix will be reduced to a half of the original as shown in (5) compare to (4).

$$I_{LF} = \begin{bmatrix} \mathbf{324} & 325 & \cdots & 340.5 & \mathbf{283.5} \\ 325 & 324 & & 339.5 & 285 \\ \vdots & & \ddots & \vdots & \\ 92 & 99 & \cdots & 200 & 194 \\ \mathbf{90.5} & 105 & & 206 & \mathbf{214.5} \end{bmatrix}_{256\times 256} \quad (5)$$

4) Result for performing zero padding to the inverse DCT

By performing a blank cell with size 2 × 2 for the whole elements of concatenated matrix, (6) will be delivered.

$$I_{IDCT+zero} = \begin{bmatrix} \mathbf{162} & 0 & \cdots & \mathbf{141.75} & 0 \\ 0 & 0 & & 0 & 0 \\ \vdots & & \ddots & \vdots & \\ \mathbf{45.25} & 0 & \cdots & \mathbf{107.25} & 0 \\ 0 & 0 & & 0 & 0 \end{bmatrix}_{512\times 512} \quad (6)$$

5) Result for obtaining the down-sampled version of the original

To obtain Figure 10 and (7), compute DCT values of (6) by exploiting the 2×2 block matrix.

Figure 10. Down-sampled version of the original image with size 256×256

$$I_{resizing} = \begin{bmatrix} \mathbf{162} & 162.5 & \cdots & 170.25 & \mathbf{141.75} \\ 162.5 & 162 & & 169.75 & 142.5 \\ \vdots & & \ddots & \vdots & \\ 46 & 49.5 & \cdots & 100 & 97 \\ \mathbf{45.25} & 52.5 & & 103 & \mathbf{107.25} \end{bmatrix}_{256\times 256} \quad (7)$$

6) Result for performing up-sampling to the down-sampled image

The up-sampled version of the resized image as shown in Figure 11 and (8) will be obtained by adding zero values to each matrix element of (7) using 2×2 block matrix scheme, followed by performing the inverse matrix.

$$I_{upsampling} = \begin{bmatrix} 162 & 162 & \cdots & 142 & 142 \\ 162 & 162 & & 142 & 142 \\ \vdots & & \ddots & \vdots & \\ 45 & 45 & \cdots & 107 & 107 \\ 45 & 45 & & 107 & 107 \end{bmatrix}_{512\times 512} \quad (8)$$

7) Result for measuring performance

By performing algorithm as shown in Figure 8 with (2) as the original and (8) as the up-sampled, the following result will be delivered.

$$The\ PSNR\ value\ is\ 35.24\ dB$$

Figure 11. Up-sampled version of the downsized image with size 512×512

Implementation of the proposed algorithm to the original Lena image (Figure 9), resulting in two images, a down-sampled version of the original image (Figure 10) and an up-sample version of the resized image (Figure 11). It is very hard to visually assess Figure 9 and Figure 11 due to human eyes limitation. In this context, an objective tool such as PSNR is required to assess both images with algorithm shown in Figure 8. Comparison of the result with literatures is listed in Table I.

TABLE I. COMPARISON OF THE PSNR FOR LENA IMAGE

Down-sizing Method			PSNR (dB)
	Up-scaling		
Down-scaling	Conventional Approach	Smart Approach	
Bilinear interpolation	Bilinear interpolation	-	34.65 [22]
Nearest neighbor	Nearest neighbor	-	34.11 [22]
DCT	Bicubic	-	35.14 [23]
DCT	Zero padding	-	35.24 [Proposed]
DCT	-	Hybrid WF	35.44 [23]
DCT	-	Adaptive *k*-NN	36.45 [23]
DCT	-	ANR-DCT	36.63 [23]
DCT	-	Deep Neural Network	36.89 [23]

A. Image Quality

Table I indicates that the proposed technique is better than bilinear and nearest neighbour interpolation in term of PSNR value. It is also compared closely to the method which combines DCT for down-scaling and bicubic for up-scaling, and with Hybrid WF method. However, if compare to the methods which exploit learning-based algorithm in determining the up-scaling image, the proposed PSNR is not as good as them. This makes sense as the proposed method is just using zero padding which means adding zero values to the empty pixel of down-sized image while the learning approach is using the smart way. This means, the proposed algorithm is still good in term of a trade-off between the quality and the cost. While for performing the proposed method to the other original images as shown in Table II, the proposed shows its strength compare to the conventional. In case using original images with different resolution, the results are shown in Table III with conclusion higher original image resolution will create higher resized image quality in term of PSNR.

B. Computational Complexity

Computational complexity is determined by number of addition, multiplication, and additional operation of the method refers to [25]. All the works are listed in Table IV which shows lowest number of substitution and multiplication than others but higher number of addition compare to nearest neighbour interpolation and scaling factor [6] but doing substitution and multiplication are more complex than addition so it can be said that the proposed method has lower complexity than both method in overall operation. While compare to the rest as listed in Table IV, the proposed is better than others [3], [6]. In order to make the trade-off more visible for the reader, the following Figure 12 shows the curve plot between the quality in term of PSNR versus the complexity in term of number of operation.

TABLE II. COMPARISON OF THE PSNR FOR OTHER ORIGINAL IMAGES WITH SIZE 512 × 512

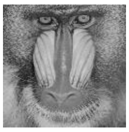
No.	Original Images	PSNR (dB)	
		Other Method	Proposed Method
1	Baboon 	21.64 [24]	24.89
2	Barbara 	25.37 [24]	28.09
3	Boat 	29.11 [24]	31.21

TABLE III. COMPARISON OF THE PSNR FOR OTHER ORIGINAL IMAGES WITH DIFFERENT SIZE OF PIXEL USING PROPOSED METHOD

No.	Original Images	PSNR (dB)
1	Mug 512 × 512 	36.93
2	Mug 1024 × 1024 	39.98

TABLE IV. COMPARISON OF COMPUTATIONAL COMPLEXITY

Method	Ref.	Operation		
		Substitution	Multiplication	Addition
Bilinear interpolation	[3]	$\odot(n^2)$	$\odot(n^3)$	$\odot(n^2)$
	[6]	1	32	24
Nearest neighbour interpolation	[3]	$\odot(n^2)$	$\odot(n^3)$	$\odot(n^2)$
	[6]	9	0	0
Scaling factor	[6]	4	5	5
Proposed		0	0	16

$\odot$=quantum operator, n=ancillary qubits

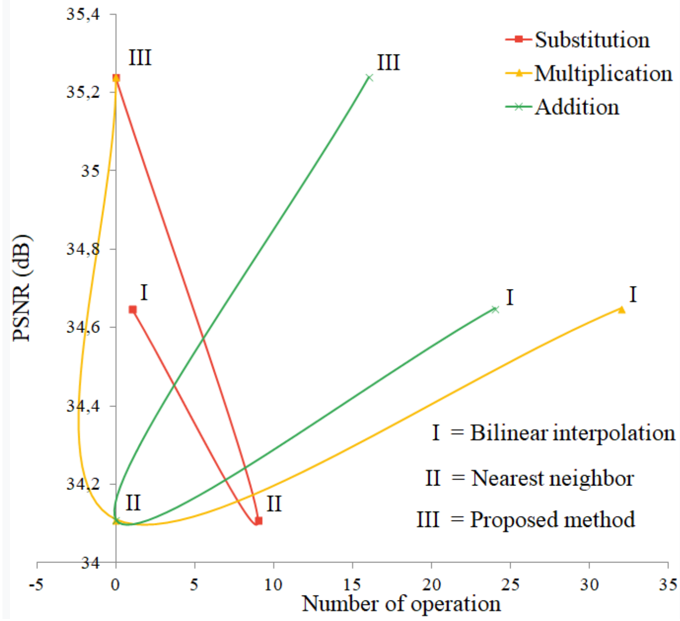


Figure 12. The quality (PSNR) versus the complexity (number of operation)

IV. CONCLUSION

Trade-off between image quality and computational complexity is possible to be achieved simply by exploiting the zero padding to the proposed algorithm. However, it will deliver lower quality of image than learning-base method in term of PSNR value. This means the reader are suggested to find the way to replace the zero padding approach by using other non smart approach but if this the option the reader still have to deal with computational complexity in term of number of operation. The clue is the reader should try to find a better balance between the quality and the complexity instead of just trying to obtain the best value of one of them. Discussion about computational complexity in term of time computation is also open for future work.

REFERENCES

[1] Y. Wang, C. Shi, C. Wang, and B. Xiao, "Measure for the Difference Between LBP Features Extracted From Original and Resized Cloud Images With Varying Resolutions," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 7, pp. 1106-1110, 2017, doi: 10.1109/LGRS.2017.2698501.

[2] K.-L. Chung, T.-Y. Liu, and J.-S. Cheng, "Novel and Optimal Luma Modification-Based Chroma Downsampling for Bayer Color Filter Array Images," *IEEE Open J. Circuits Syst.*, vol. 1, no. 1, pp. 48-59, 2020, doi: 10.1109/ojcas.2020.2996624.

[3] F. Yan, S. Zhao, S. E. Venegas-Andraca, and K. Hirota, "Implementing bilinear interpolation with quantum images," *Digit. Signal Process. A Rev. J.*, vol. 117, no. 1, pp. 103149-103161, 2021, doi: 10.1016/j.dsp.2021.103149.

[4] R. G. Zhou and C. Wan, "Quantum Image Scaling Based on Bilinear Interpolation with Decimals Scaling Ratio," *Int. J. Theor. Phys.*, vol. 60, no. 6, pp. 2115-2144, 2021, doi: 10.1007/s10773-021-04829-6.

[5] G. Sheng, T. Li, Q. Su, B. Chen, and Y. Tang, "Detection of content-aware image resizing based on Benford's law," *Soft Comput.*, vol. 21, no. 19, pp. 5693-5701, 2017, doi: 10.1007/s00500-016-2146-6.

[6] R.-G. Zhou, Y. Cheng, X. Qi, H. Yu, and N. Jiang, "Asymmetric scaling scheme over the two dimensions of a quantum image," *Quantum Information Processing*, vol. 19, no. 9, pp. 343-362, 2020, doi: 10.1007/s11128-020-02837-9.

[7] Y. W. Kim, J. Innila Rose, and A. V. N. Krishna, "A Study on the Effect of Canny Edge Detection on Downscaled Images," *Pattern Recognit. Image Anal.*, vol. 30, no. 3, pp. 372-381, 2020, doi: 10.1134/S1054661820030116.

[8] N. Weber, M. Waechter, S. C. Amend, S. Guthe, and M. Goesele, "Rapid, detail-preserving image downscaling," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 205-210, 2016, doi: 10.1145/2980179.2980239.

[9] M. Zhang, T. Chen, X. Shi, and P. Cao, "Image arbitrary-ratio down- and up-sampling scheme exploiting dct low frequency components and sparsity in high frequency components," *IEICE Trans. Inf. Syst.*, vol. E99-D, no. 2, pp. 475-487, 2016, doi: 10.1587/transinf.2015EDP7141.

[10] E. S. L. Gastal and M. M. Oliveira, "Spectral remapping for image downscaling," *ACM Trans. on Graph*, vol. 36, no. 4, pp. 145-160, 2017. doi: 10.1145/3072959.3073670.

[11] M. Xiao *et al.*, "Invertible Image Rescaling," *Eur. Conf. on Comput. Vision*, vol. 12346, no. 1, pp. 126-144, 2020. doi: 10.1007/978-3-030-58452-8_8.

[12] S. I. Cho and S. J. Kang, "Learning Methodologies to Generate Kernel-Learning-Based Image Downscaler for Arbitrary Scaling Factors," *IEEE Trans. Image Process.*, vol. 30, no. 1, pp. 4526-4539, 2021, doi: 10.1109/TIP.2021.3073316.

[13] W. Sun and Z. Chen, "Learned Image Downscaling for Upscaling Using Content Adaptive Resampler," *IEEE Trans. Image Process.*, vol. 29, no. 1, pp. 4027-4040, 2020, doi: 10.1109/TIP.2020.2970248.

[14] C. Yao, J. Xiao, T. Tillo, Y. Zhao, C. Lin, and H. Bai, "Depth Map Down-Sampling and Coding Based on Synthesized View Distortion," *IEEE Trans. Multimed.*, vol. 18, no. 10, pp. 2015-2022, 2016, doi: 10.1109/TMM.2016.2594145.

[15] S. Wang, K. Gu, S. Ma, and W. Gao, "Joint Chroma Downsampling and Upsampling for Screen Content Image," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 9, pp. 1595-1609, 2016, doi: 10.1109/TCSVT.2015.2461891.

[16] S. E. Tandogan, E. Altinişik, S. Sarimurat, and H. T. Sencar, "Tackling in-camera downsizing for reliable camera ID verification," SIST Electron. Imag., vol. 31, no. 1, pp. 545-551, 2019. doi: 10.2352/ISSN.2470-1173.2019.5.MWSF-545.

[17] I. Bouderbal, A. Amamra, and M. A. Benatia, "How Would Image Down-Sampling and Compression Impact Object Detection in the Context of Self-driving Vehicles ?," *Int. Conf. on Comput. Syst. and Appl.*, vol. 199, no. 1, pp. 25-37, 2021. doi: 10.1007/978-3-030-69418-0_3.

[18] J. Liu, S. He, and R. W. H. Lau, "L0-regularized image downscaling," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1076-1085, 2018, doi: 10.1109/TIP.2017.2772838.

[19] J. J. García Aranda, M. A. Granero, F. J. J. Quintanilla, G. Caffarena, and R. García-Carmona, "Elastic downsampling: An adaptive downsampling technique to preserve image quality," *Electron.*, vol. 10, no. 4, pp. 400-419, 2021, doi: 10.3390/electronics10040400.

[20] R. Gupta, D. Mehrotra, and R. K. Tyagi, "Computational complexity of fractal image compression algorithm," *IET Image Process.*, vol. 14, no. 17, pp. 4425-4434, 2020, doi: 10.1049/iet-ipr.2019.0489.

[21] I. Setiawan, A. I. Basuki, and D. Rosiyadi, "Optimized Hybrid DCT-SVD Computation over Extremely Large Images," *J. Tek. Elektro*, vol. 13, no. 2, pp. 56-61, 2021, doi: 10.15294/jte.v13i2.31879.

[22] G. Chen, H. Zhao, C. K. Pang, T. Li, and C. Pang, "Image Scaling: How Hard Can it Be?," *IEEE Access*, vol. 7, no. 1, pp. 129452-129465, 2019, doi: 10.1109/ACCESS.2019.2940353.

[23] K. W. Hung, K. Wang, and J. Jiang, "Image up-sampling using deep cascaded neural networks in dual domains for images down-sampled in DCT domain," *J. Vis. Commun. Image Represent.*, vol. 56, no. 1, pp. 144-149, 2018, doi: 10.1016/j.jvcir.2018.09.005.

[24] J. McAvoy, E. Rahimi, and C. Joslin, "Low Complex Image Resizing Algorithm using Fixed-point Integer Transformation," *Int. Conf. on Computer Vision, Imaging and Computer Graphics Theory and Applications.*, vol. 4, no. 1, pp. 143-149, 2018. doi: 10.5220/0006616901430149.

[25] N. Brahimi, T. Bouden, T. Brahimi, and L. Boubchir, "A novel and efficient 8-point DCT approximation for image compression," *Multimed. Tools Appl.*, vol. 79, no. 2, pp. 7615-7631, 2020, doi: 10.1007/s11042-019-08325-2.