Security of Infocommunication Systems and Internet of Things

2024 Vol 2, No 1

https://doi.org/10.31861/sisiot2024.1.01007

Received 18 July 2024; revised 12 August 2024; accepted 22 August 2024; published 30 August 2024

Analysis of Results of Scaling Digital Images by Interpolation Algorithms

Serhiy Balovsyak[1,*] and Yurii Hnatiuk[2]

[1]Computer Systems and Networks Department, Yuriy Fedkovych Chernivtsi National University, Chernivtsi, Ukraine
[2]Radio Engineering and Information Security Department, Yuriy Fedkovych Chernivtsi National University, Chernivtsi, Ukraine

*Corresponding author (E-mail: s.balovsyak@chnu.edu.ua)

ABSTRACT Scaling of digital bitmap images is often used in modern computer and telecommunications systems. Among image scaling algorithms, the most common are interpolation algorithms, namely nearest neighbor, Lanczos, bilinear and bicubic interpolations. However, in the process of scaling images by interpolation algorithms, characteristic distortions appear on them. Therefore, in this work, software implementation and research of image interpolation algorithms was performed in order to identify their advantages and disadvantages, areas of use and ways of improvement. In order to study the interpolation algorithms, a scaled fRGBs image was calculated based on the initial image fRGB, and then a scaled image fRGBs2 with the dimensions of the original image was calculated based on fRGBs. The scaling accuracy is evaluated using the Root Mean Square Error between the pixel values of the original and scaled images. The software for scaling images is developed in the Python language. Calculation of scaled images is performed by the cv2.resize() function of the OpenCV library. Using the developed program, scaling of a series of 100 images was carried out, the accuracy and speed of digital image scaling by interpolation algorithms were investigated. For each interpolation method, the average scaling error AR and the limits of its confidence interval ARmin and ARmax with a specified reliability γ are calculated. The average scaling time AT of a series of 100 images by different algorithms was determined. The research results showed that in most cases the smallest scaling error is provided by the bicubic interpolation algorithm, which is slightly inferior in speed to the nearest neighbor and bilinear interpolation algorithms. Recommendations for the application of interpolation algorithms have been developed. It is shown that the use of convolutional neural networks is promising for the highest quality image scaling.

KEYWORDS scaling of digital images, nearest neighbor algorithm, bilinear Interpolation, bicubic interpolation, Lanczos algorithm.

I. INTRODUCTION

Scaling of digital raster images is an important task in modern computer and telecommunication systems [1, 2]. Reducing or increasing the scale of images is done by reducing or increasing their resolution. Scaling down of images is used, for example, to reduce the load on the channels of the telecommunications system and to reduce the amount of memory for saving images. Scaling up of images is used, for example, to restore the original dimensions of images and to improve their further computer processing [3]. From the point of view of digital signal processing, the scaling of raster images is a two-dimensional example of the conversion of the sampling rate for a discrete signal [4].

Among digital image scaling algorithms, the most common are interpolation algorithms, which are characterized by simplicity of software or hardware implementation, high speed, and relatively high visual quality of scaled images [5]. The following image interpolation algorithms are common: nearest neighbor, Lanczos, bilinear and bicubic interpolations [2, 6]. Other methods of image scaling, in particular, using the Fourier transform [1], multiscale processing [7] and artificial neural networks [8, 9], are characterized by more complex implementation and lower speed.

The problem is that in the process of scaling images by interpolation algorithms, characteristic distortions (artifacts) appear on them, contours are blurred, image detail is lost. It is important that different interpolation algorithms distort the image in different ways and have different speeds. Therefore, the goal of the work is relevant, which consists in the software implementation and research of image interpolation algorithms in order to identify their advantages and disadvantages, areas of use and ways of improvement.

II. THEORETICAL BASICS OF IMAGE INTERPOLATION ALGORITHMS

Interpolation algorithms are used for scaling (resizing) images. Digital color images are processed as $f_{RGB}(i, k, c)$ arrays, where $i = 0,..., M-1$; $k = 0,..., N-1$; $c = 0,..., Q_C-1$; M is the number of image pixels by height, N is the number of pixels by width; $Q_C = 3$ is the number of color channels (Red, Green, Blue) [1, 2]. The values of all color channels are normalized in the range from 0 to 1.

Let's consider the principles of the following interpolation algorithms: nearest neighbor, bilinear and bicubic interpolations, Lanczos.

The nearest neighbor algorithm is the simplest. The principle of its operation boils down to the selection of one of the pixel values that are closest to the new pixel position after resizing the image.

The bilinear image interpolation algorithm is that the new pixel value is calculated using a linear combination of the values of the 4 pixels that are closest to the new pixel position. In the case of image scaling, linear interpolation is performed first in one direction (height) and then in another (width).

The bicubic image interpolation algorithm is that the new pixel value is calculated using the cubic interpolation of the pixel values in two coordinates (height and width). In the case of cubic interpolation, the value of the function (pixel brightness) at the desired point is calculated from its value at 16 neighboring points.

The Lanczos algorithm uses the Lanczos window function (kernel) to calculate a new pixel value based on the values of neighboring pixels when the image is resized. In the case of interpolation of one-dimensional signals $s(x)$, the function $\mathrm{sinc}(x/a)$ is used as the Lanczos kernel, where $-a \le x \le a$. The two-dimensional Lanczos kernel is used in image processing. The parameter a is a positive integer (usually $a = 3$) that defines the size of the kernel.

In order to study interpolation algorithms based on the initial image f_{RGB}, the calculation of the scaled image f_{RGBs} with the size of $M_s \times N_s$ pixels (with the scaling factor S_c, for example, $S_c = 0.5$) was performed, and then, based on f_{RGBs}, the calculation of the scaled image f_{RGBs2} with the dimensions of the initial image of $M \times N$ pixels is performed.

Ideally, the original f_{RGB} and the scaled f_{RGBs2} images should match. However, due to scaling, there are differences between the images. Scaling error are calculated from the difference between the f_{RGB} and f_{RGBs2} images as the Root Mean Square Error (RMSE) using the equation:

$$R_{MSE} = \sqrt{\frac{1}{MN}\sum_{i=0}^{M-1}\sum_{k=0}^{N-1}\left[f_{RGB}(i,k) - f_{RGBs2}(i,k)\right]^2} \, . \quad (1)$$

The average value of RMSE for a series of Q_{Im} ($Q_{Im} \ge 10$) images is denoted by A_R, and the standard deviation of RMSE values for a series of Q_{Im} images is denoted by σ_R. Testing of the hypothesis of a normal distribution of RMSE values is performed using the Pearson's χ^2 test [10]. The hypothesis of a normal distribution is confirmed if $\chi^2_R < \chi^2_T(\alpha, k)$, where χ^2_R is the observed value of the Pearson test, χ^2_T is the critical point of the right-hand region for the χ^2 distribution, α is the significance level, $k = (Q_h - 3)$ is quantity of the degrees of freedom, Q_h is the quantity of histogram intervals. The selection of the optimal Q_h is performed according to the Sturges' rule [11]

$$Q_h = \log_2(Q_{\mathrm{Im}}) + 1 , \quad (2)$$

where Q_{Im} is the number of objects (images).

If the hypothesis of a normal RMSE distribution is confirmed, it is possible to construct a confidence interval $A_{Rmin} < A_R < A_{Rmax}$ for the average scaling error A_R with a specified reliability γ [10]. The lower limit A_{Rmin} of the confidence interval is calculated by the formula

$$A_{R\min} = A_R - \frac{x_L \cdot \sigma_R}{\sqrt{Q_{\mathrm{Im}}}} , \quad (3)$$

where x_L is the value of the Laplace function for which $\Phi(x_L)=\gamma/2$. Similarly, the upper limit A_{Rmax} of the confidence interval is calculated using the formula

$$A_{R\max} = A_R + \frac{x_L \cdot \sigma_R}{\sqrt{Q_{\mathrm{Im}}}} . \quad (4)$$

Thus, for each interpolation method, it is possible to scale a series of images and calculate not only the average scaling error A_R, but also the limits A_{Rmin} and A_{Rmax} of its confidence interval.

III. SOFTWARE IMPLEMENTATION OF IMAGE INTERPOLATION ALGORITHMS

The image scaling software was developed in Python using OpenCV (cv2), Numpy, Matplotlib, etc. libraries. Calculation of scaled images is performed by the cv2.resize() function of the OpenCV library using the following interpolation methods:

- cv2.INTER_NEAREST is a method based on the nearest neighbor algorithm, which is characterized by simplicity and high speed, but can lead to loss of details in the image.
- cv2.INTER_LINEAR is a method based on the linear interpolation algorithm. The method usually gives better results (compared to INTER_NEAREST), but it smoothes the image to some extent.
- cv2.INTER_CUBIC is a method based on the cubic interpolation algorithm. The method usually gives better results (compared to INTER_LINEAR), but is slower.
- cv2.INTER_LANCZOS4 is a method based on the Lanczos algorithm, which has a relatively low speed (compared to the cv2.INTER_CUBIC method). Using this method allows to get scaled images with high accuracy, but the appearance of unwanted artifacts such as ringing is possible.

IV. ANALYSIS OF RESULTS FOR IMAGE INTERPOLATION ALGORITHMS

Using the developed software, a series of Q_{Im} images was read from graphic files and scaled (Fig. 1).

On the basis of each initial f_{RGB} image (Fig. 2), a scaled image f_{RGBs} with the size of $M_s \times N_s$ pixels with a scaling factor of $S_c = 0.5$ ($M_s = M \cdot S_c$, $N_s = N \cdot S_c$) was calculated using the bicubic interpolation algorithm (Fig. 3a). Next, based on f_{RGBs}, a scaled image f_{RGBs2} with the dimensions of the initial image $M \times N$ pixels was calculated (Fig. 3b). Similarly, image scaling was performed using the bilinear interpolation, Lanczos and nearest neighbor algorithms.

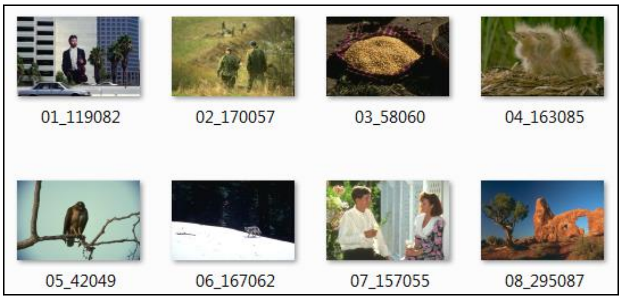
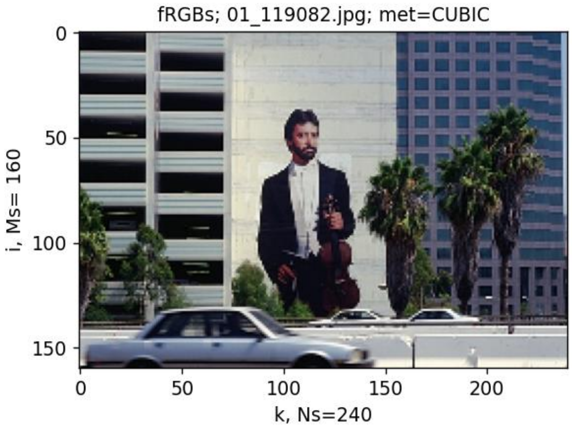
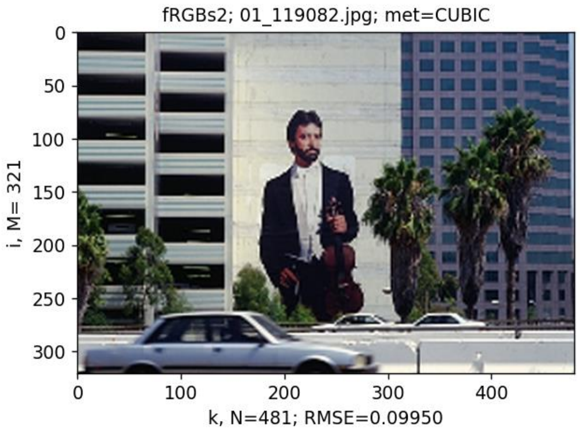


FIG. 1. Image base BSDS300 [12, 13], 8 images from 100 are shown.

FIG. 2. Initial image f_{RGB} #01.



a)



b)

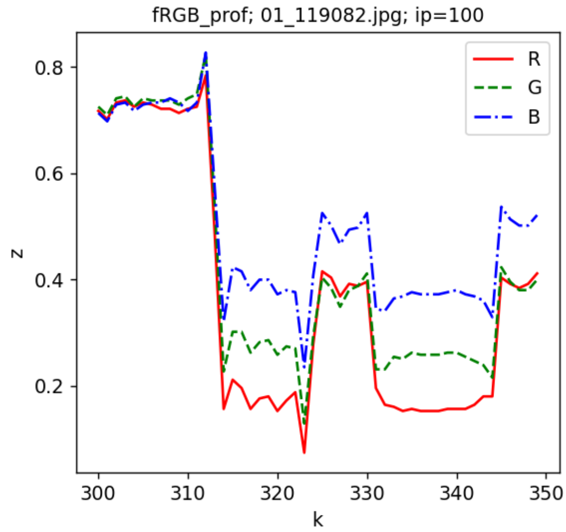
FIG. 3. Image f_{RGBs} after scaling down (a) and image f_{RGBs2} after scaling up (b) using the 'CUBIC' method.

Distortions that occur when scaling images are especially noticeable on their profiles (Fig. 4).

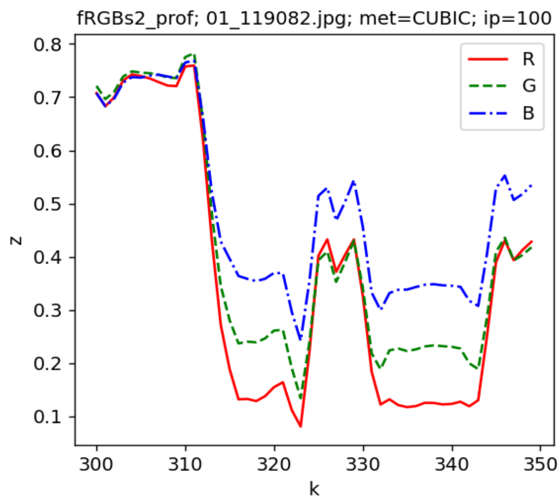
For each scaled image f_{RGBs2}, a certain value of scaling error RMSE (1) is obtained (Fig. 5).

For most images, the scaling error RMSE by the 'LINEAR' method is greater than by the 'CUBIC' method (Fig. 6); the scaling error by the 'LANCZOS4' method is greater than by the 'LINEAR' method (Fig. 7); the scaling error by the 'NEAREST' method significantly exceeds the errors of other methods (Table I).

The test of the hypothesis of a normal distribution of RMSE values was performed according to the Pearson's χ^2 test for the significance level $\alpha = 0.05$. The optimal quantity of intervals of the histogram $Q_h = 8$ is calculated according to formula (2), so the quantity of the degrees of freedom is $k = 5$. According to the tabular data, the critical point is $\chi^2_T(0.05, 5) = 11.07$. For the 'CUBIC', 'LINEAR', 'LANCZOS4', 'NEAREST' methods, the observed values of the Pearson's χ^2_R test were obtained,



a)



b)

FIG. 4. Fragments of the RGB profiles $z(k)$ of the initial image f_{RGB} (a) and the 'CUBIC' method scaled image f_{RGBs2} (b) for the red, green and blue components; i_p is the pixel row number.

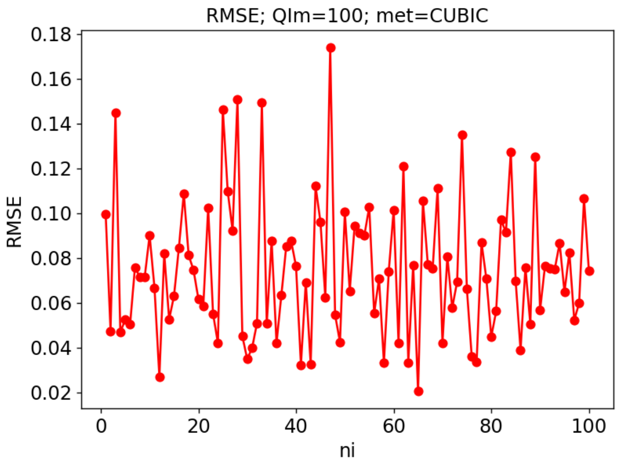


FIG. 5. Values of scaling error RMSE for $Q_{Im} = 100$ images (Fig. 1), scaled by the 'CUBIC' method; n_i is image numbers.

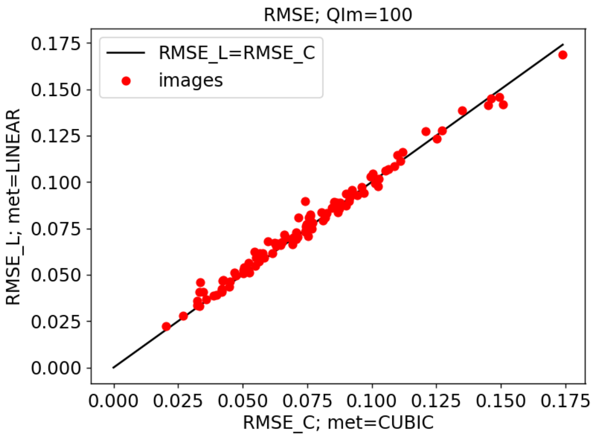


FIG. 6. 3 Dependence of the scaling error RMSE by the 'LINEAR' method (RMSE_L) from the scaling error by the 'CUBIC' method (RMSE_C) for a series of 100 images (Fig. 1).

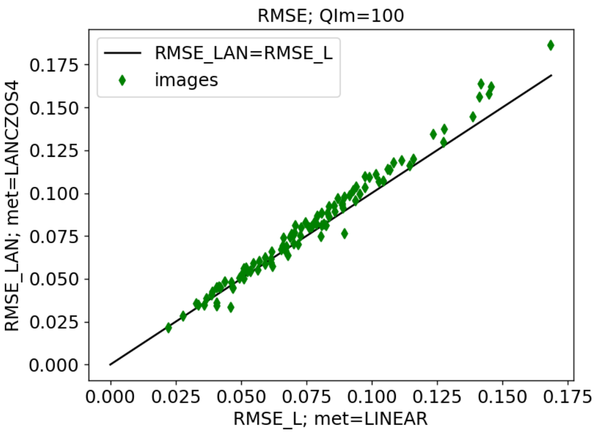


FIG. 7. Dependence of the scaling error RMSE by the 'LANCZOS4' method (RMSE_LAN) from the scaling error by the 'LINEAR' method (RMSE_L) for a series of 100 images (Fig. 1).

which are 9.661, 07.449, 9.370 and 5.643. The hypothesis of a normal distribution of RMSE values is confirmed for all methods, since for all methods χ2R < χ2T.

For each interpolation method, the average scaling error AR and the standard deviation σR for the RMSE values were calculated, and the limits of the confidence interval ARmin < AR < ARmax were calculated using formulas (3), (4) with a reliability of γ = 0.99 (Table 1, Fig. 8).

Thus, based on the average scaling error A_R and taking into account the limits A_{Rmin} and A_{Rmax} of its confidence interval, the smallest scaling error is provided by the 'CUBIC' method, a slightly higher error by the 'LINEAR' method, an even larger error by the 'LANCZOS4' method, and the largest scaling error by the ' NEAREST' method.

TABLE 1. Average scaling error A_R and standard deviation σ_R for the RMSE values, limits of the confidence intervals $A_{Rmin} < A_R < A_{Rmax}$ for a series of Q_{Im} = 100 images (Fig. 1).

	CUBIC	LINEAR	LANCZOS4	NEAREST
A_R	0.075024	0.076572	0.080177	0.177774
σ_R	0.030376	0.029270	0.032890	0.057526
A_{Rmin}	0.067186	0.069020	0.071692	0.162933
A_{Rmax}	0.082861	0.084124	0.088663	0.192616

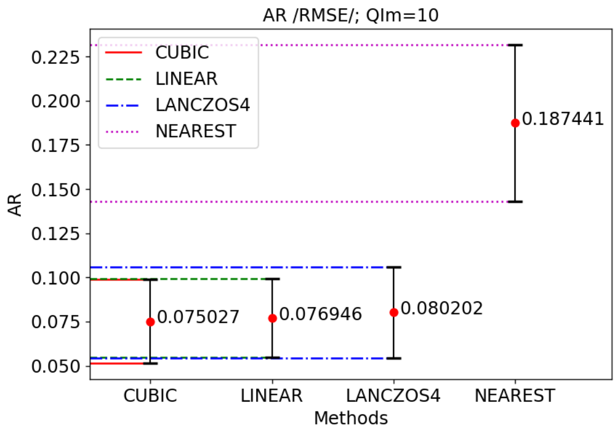


FIG. 8. Average scaling errors A_R and limits of its confidence intervals for Q_{Im} = 100 images (Fig. 1).

With the 'LINEAR' method, the best scaling results are obtained for images with smooth changes in brightness and without clear contours. The Lanczos method allows to preserve the sharpness of edges and detail, which is useful for scaling images with a lot of detail (text, high-contrast graphic elements).

To compare the speed of the methods, the average scaling time A_T was determined, during which the scaling of a series of Q_{Im} images (Fig. 1) is performed by different methods (Table 2, Fig. 9). A computer with an AMD A4-6300 Processor, 3.70 GHz was used. The limits of the confidence intervals $A_{Tmin} < A_T < A_{Tmax}$ with reliability γ = 0.99 were also calculated.

In the process of determining the time of image interpolation, the scaling factor S_c = 8 was used, so the total time of image scaling turned out to be significant. Due to this, it was possible to clearly divide the confidence intervals for the time of interpolation of images by different methods (Fig. 9). It was found that when scaling a series of investigated images, the smallest processing time was obtained for the 'LINEAR' method, and the largest time for the 'LANCZOS4' method.

Research results showed that bilinear interpolation algorithm has the highest speed, but in terms of accuracy it is slightly inferior to the bicubic interpolation algorithm due to partial image smoothing. Therefore, the bicubic interpolation algorithm can be used in cases where maximum scaling speed is required, or for processing images with smooth changes in brightness and without clear contours.

The Lanczos algorithm preserves edge sharpness and detail, which is useful for scaling images with a lot of detail. However, the Lanczos algorithm has the lowest speed, so it is not recommended to use it in high-speed systems.

TABLE 2. The average scaling time A_T for a series of Q_{Im} = 100 images and the limits of the confidence intervals $A_{Tmin} < A_T < A_{Tmax}$; image scaling factor S_c = 8.

	CUBIC	LINEAR	LANCZOS4	NEAREST
A_T, s	22.13042	16.34675	47.48684	19.19302
A_{Tmin}, s	21.71049	16.04037	46.85289	18.95186
A_{Tmax}, s	22.55036	16.65312	48.12078	19.43418

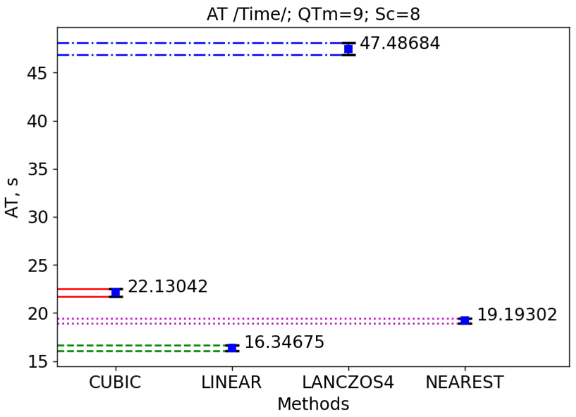


FIG. 9. Average scaling time A_T for a series of Q_{Im} = 100 images and limits of its confidence intervals; quantity of scaling attempts Q_{Tm} = 9, image scaling factor S_c = 8.

The nearest neighbor algorithm has a significantly higher interpolation error compared to other algorithms, so this method is useful only for comparison or training purposes.

V. CONCLUSION

A study of the accuracy and speed of digital image scaling using interpolation algorithms was conducted. Algorithms of nearest neighbor, bilinear and bicubic interpolations, Lanczos are considered. Scaling accuracy is estimated through the RMSE between the pixel values of the original and scaled images.

The main scientific contribution of the authors is that, according to the Pearson's χ^2 test, the hypothesis of a normal distribution of RMSE values for all investigated image interpolation algorithms is confirmed. Due to this, for each interpolation method, not only its average scaling error A_R is calculated, but also the bounds of its confidence interval A_{Rmin} and A_{Rmax} with a given reliability γ. Such an analysis of interpolation algorithms made it possible to more accurately assess their accuracy and speed, as well as advantages and disadvantages, areas of use and ways of improvement.

The research results showed that in most cases the smallest scaling error is provided by the bicubic interpolation algorithm, which is slightly inferior in speed to the nearest neighbor and bilinear interpolation algorithms. Therefore, in the general case, it is advisable to scale images using the bicubic interpolation algorithm.

The application of the considered interpolation algorithms leads to certain image distortions. Therefore, the use of artificial neural networks [14], in particular, convolutional neural networks, is promising for the highest quality image scaling.

AUTHOR CONTRIBUTIONS

S.B. – conceptualization, writing-review and editing, supervision; Yu.H – methodology, software, resources, writing-original draft preparation, visualization, validation, investigation.

COMPETING INTERESTS

The authors declare no conflict of interest.

REFERENCES

[1] R. Gonzalez and R. Woods, *Digital Image Processing*, 4th ed. Pearson/Prentice Hall, NY, 2018.

[2] S. Palani, *Principles of Digital Signal Processing*. Springer Cham, 2022.

[3] D. E. Kim and D. Hwang, Eds., *Intelligent Imaging and Analysis*. MDPI, Basel, Switzerland, 2020.

[4] S. Balovsyak, V. Lacusta, and K. Odaiska, "Reading of Sensor Signals with Automatic Selection of Sampling Frequency," *Security of Infocommunication Systems and Internet of Things*, vol. 1, no. 1, paper 01010, pp. 1-4, 2023. doi: 10.31861/sisiot2023.1.01010.

[5] D. Suresha and H. N. Prakash, "Data Content Weighing for Subjective versus Objective Picture Quality Assessment of Natural Pictures," *Int. Journal of Image, Graphics and Signal Processing (IJIGSP)*, vol. 9, no. 2, pp. 27-36, 2017.

[6] G. Liu, "The Novel Bilateral Quadratic Interpolation Image Super-resolution Algorithm," *International Journal of Image, Graphics and Signal Processing (IJIGSP)*, vol. 13, no. 3, pp. 55-61, 2021. doi: 10.5815/ijigsp.2021.03.05.

[7] P. Prystavka and O. Cholyshkina, "Pyramid Image and Resize Based on Spline Model," *International Journal of Image, Graphics and Signal Processing (IJIGSP)*, vol. 14, no. 1, pp. 1-14, 2022. doi: 10.5815/ijigsp.2022.01.01.

[8] S. Balovsyak, I. Fodchuk, K. Odaiska, Yu. Roman, and E. Zaitseva, "Analysis of X-Ray Moiré Images Using Artificial Neural Networks," in *IntelITSIS 2022: 3rd Int. Workshop on Intelligent Information Technologies and Systems of Information Security*, March 23–25, 2022, Khmelnytskyi, Ukraine, CEUR Workshop Proceedings, pp. 187-197, 2022.

[9] Waifu2x [Online]. Available: https://waifu2x.udp.jp/index.uk.html.

[10] O. Knill, *Probability and Stochastic Processes with Applications*. Overseas Press (India), Pvt. Ltd, 2009.

[11] D. W. Scott, "Sturges' Rule," *WIREs: Computational Statistics*, vol. 1, no. 3, pp. 303-306, 2009.

[12] C. Fowlkes, D. Martin, and J. Malik, "Local Figure/Ground Cues are Valid for Natural Images," *Journal of Vision*, vol. 7, no. 8, pp. 1-9, 2007.

[13] The Berkeley Segmentation Dataset and Benchmark, BSDS300 [Online]. Available: https://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds.

[14] A. Geron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, Inc., 2019.

Serhiy Balovsyak

In 1995, graduated from Chernivtsi State University. In 2018, defended his doctoral dissertation in the specialty "Computer systems and components". Currently, works as an associate professor at the Department of Computer Systems and Networks of Chernivtsi National University. Research interests include digital signal processing, programming, artificial neural networks.

ORCID ID: 0000-0002-3253-9006

Yurii Hnatiuk

In 2019, graduated from Chernivtsi National University with a degree in "Computer Engineering". In 2002, entered postgraduate studies in the specialty "Telecommunications and radio engineering". Research interests include digital signal and image processing, programming, artificial neural networks.

ORCID ID: 0009-0006-5605-9281

Аналіз результатів масштабування цифрових зображень алгоритмами інтерполяції

Сергій Баловсяк[1,*], Юрій Гнатюк[2]

[1]Кафедра комп'ютерних систем та мереж, Чернівецький національний університет імені Юрія Федьковича, Чернівці, Україна
[2]Кафедра радіотехніки та інформаційної безпеки, Чернівецький національний університет імені Юрія Федьковича, Чернівці, Україна

*Автор-кореспондент (Електронна адреса: s.balovsyak@chnu.edu.ua)

АНОТАЦІЯ Масштабування цифрових растрових зображень часто використовується в сучасних комп'ютерних і телекомунікаційних системах. Серед алгоритмів масштабування зображень найбільш поширеними є алгоритми інтерполяції, а саме найближчого сусіда, Ланцоша, білінійної та бікубічної інтерполяцій. Проте, в процесі масштабування зображень алгоритмами інтерполяції на них виникають характерні спотворення. Тому в даній роботі виконано програмну реалізацію та дослідження алгоритмів інтерполяції зображень з метою виявлення їх переваг і недоліків, сфер використання та шляхів удосконалення. З метою дослідження алгоритмів інтерполяції на основі початкового зображення fRGB виконано обчислення масштабованого зображення fRGBs, а потім на основі fRGBs обчислено масштабоване зображення fRGBs2 із розмірами початкового зображення. Точність масштабування оцінено через корінь середньої квадратичної помилки (root mean square error – RMSE) між значеннями пікселів початкового та масштабованого зображень. Програму для масштабування зображень розроблено на мові Python. Обчислення масштабованих зображень виконано функцією cv2.resize() бібліотеки OpenCV. За допомогою розробленої програми проведено масштабування серії з 100 зображень, досліджено точність та швидкодію масштабування цифрових зображень за алгоритмами інтерполяції. Для кожного метода інтерполяції обчислено середню помилку масштабування AR та межі її довірчого інтервалу ARmin та ARmax із заданою надійністю γ. Визначено середній час масштабування AT для серії з 100 зображень різними алгоритмами. Результати досліджень показали, що у більшості випадків найменшу помилку масштабування забезпечує алгоритм бікубічної інтерполяції, який незначно поступається за швидкодією алгоритмам найближчого сусіда та білінійної інтерполяції. Розроблено рекомендації по застосуванню алгоритмів інтерполяції. Показано, що для найбільш якісного масштабування зображень є перспективним застосування згорткових нейронних мереж.

КЛЮЧОВІ СЛОВА масштабування цифрових зображень, алгоритм найближчого сусіда, білінійна інтерполяція, бікубічна інтерполяція, алгоритм Ланцоша.