Research of algorithms based on fractional differential derivatives for improving medical images*

Yaroslav Sokolovskyy[1,*,†], Volodymyr Bereziuk[2,†], Mariana Levkovych[1,†] and Mykhailo Paslavskyi[2,†]

[1] *Department of Computer Design Systems, Lviv Polutechnic National University, S. Bandery street 12, Lviv, Ukraine*
[2] *Computer Science Department, National Forestry University of Ukraine, Henerala Chuprynky street, 103, Lviv, Ukraine*

Abstract

This article is dedicated to the study of fractal operators and the review of their application for processing medical MRI images. The Atangana-Baleanu fractial operator is discussed in detail. Four numerical approximations of this operator are described, along with the process of finding the approximation coefficients. Based on the approximation operators, differently oriented mask are created for image denoising. These masks are implemented in the developed algorithm and software product, which allows improving image quality. A visual analysis of the results of the algorithms is performed, comparing the denoising image with original by using analytical params. Additionally, the results of the developed algorithms are compared with other algorithms, both based on fractial operators and classical image denoising methods. Conclusions regarding the use of the method based on the Atangana-Baleanu fractial operator for processing medical MRI images are presented.

Keywords

Medical images, Magnetic Resonance Imaging (MRI), fractal operators, Atangana-Baleanu derivative.

1. Introduction

Medical imaging is one of the tools used for diagnosing and monitoring patient conditions. It includes X-rays, ultrasound, computed tomography (CT), magnetic resonance imaging (MRI), and others. Each of these methods is used to visualize different body structures. MRI images provide detailed scans of internal organs and tissues, particularly the brain. Brain MRI is important for diagnosing diseases such as tumors, strokes, aneurysms, and other conditions that affect brain function. The quality of medical images can vary, which is especially important for brain MRI, where image clarity directly impacts diagnostic accuracy. Blurry, low-quality and noisy images can complicate data interpretation.

Image quality enhancement and denoising is possible through computer algorithms that help remove noise and in the same time save structures on image. Possible method involves the use of filter-based algorithms [1, 2, 3], which help emphasize contours and edges in images, which facilitates more accurate data analysis.

Fractial operators are used to model processes characterized by complex dynamics and irregular structures. This applies to phenomena where traditional mathematical methods cannot adequately describe system behavior, such as “memory” effects and self-organization in deformation-relaxation processes [4, 5], biological systems [6], and so on. Fractial operators help model processes with

IDDM'24: 7th International Conference on Informatics & Data-Driven Medicine, November 14 - 16, 2024, Birmingham, UK
* Corresponding author.
† These authors contributed equally.
yaroslav.i.sokolovskyi@lpnu.ua (Ya. Sokolovskyy); volodymy.berezyuk@nltu.edu.ua (V. Bereziuk); mariana.v.levkovych@lpnu.ua (M. Levkovych); mykhailo.paslavskyi@nltu.edu.ua (M. Paslavskyi)
0000-0003-4866-2575 (Ya. Sokolovskyy); 0009-0001-6930-4235 (V. Bereziuk); 0009-0004-6231-7799 (M. Levkovych); 0000-0003-1635-4340 (M. Paslavskyi);

nonlinear dynamics, chaotic behavior, and self-similar structures, allowing for a more accurate description of their evolution and properties across different scales. Additionally, fractal operators are used in various types of image processing, including edge detection, overall image enhancement, noise reduction, and more [7]. They serve as an effective tool for analyzing complex structures in images, such as organ tissues or other heterogeneous regions.

Noise refers to unwanted signals or disturbances that disrupt or alter the main data, appearing at various frequencies and impacting the quality of signals or images. Fractional integrals are effective in reducing high-frequency noise while maintaining low-frequency components that carry important information. By adjusting the order of the integral, these methods facilitate efficient data smoothing and enhance complex textures without compromising their structure. This allows to improve image details without losing important information about the main structures, which is especially important when analyzing medical images.

Image enhancement using fractial operators, such as the Grunwald-Letnikov and Riemann-Liouville operators, is discussed in scientific research [8, 9]. These operators are used in image processing, particularly for edge detection, noise reduction and the enhancement of important structures.

A new fractial operator is the Atangana-Baleanu operator [10], which differs from classical operators by offering greater flexibility and adaptability in processing complex images. This operator takes into account system memory, allowing to model the influence of past states on the current process. Thanks to the Mittag-Leffler kernel, the effects of past events decay more slowly, which provides more accurate simulation of processes. It has become an innovative tool in fractial processing and opens new possibilities for analysis. The Atangana-Baleanu operator has already been used for edge detection in images [11], but its potential for denoising medical MRI images has not yet been explored. This article presents a new application of this operator for denoising of brain MRI imagesNoise reduction in images can be assessed using PSNR and SIMM metrics. By utilizing numerical approximations from [12] and developing an algorithm based on fractional masks, we will evaluate the effectiveness of noise reduction in medical images. A comparison of the denoised medical images with results from other algorithms will be conducted.

The following section of the work involves the application of software-algorithmic tools to reduce noise of medical MRI images using a fractial operators. The use of this method will allow to improve quality of noised MRI images, contributing to more accurate diagnosis, better visual representation, and enhanced automated systems for recognizing and classifying pathologies in MRI images.

2. The main material is presentation

2.1. Fractial operators

Fractal operators based on fractional-order derivatives [13] are mathematical expressions used to describe signals with fractial properties. These operators include the Grünwald-Letnikov, Riemann-Liouville, Caputo, Atangana-Baleanu operators, and so on.

The fractional-order derivative based on the Grunwald-Letnikov [13] fractial differential σ is expressed by the following formula:

$$D^{\sigma}_{G-L}s(x) = \frac{d^{\sigma}}{[d(x-a)]^{\sigma}}s(x)_{G-L} = \lim_{N\to\infty}\{\frac{(\frac{x-a}{N})}{\Gamma(-\sigma)}\sum_{k=0}^{N-1}\frac{\Gamma(k-\sigma)}{\Gamma(k+1)} * s\,(x-k\,(\frac{x-a}{N}))\}, \quad (1)$$

where signal length s(x) is whithin the interval [a, x], σ -any real number (including fractional values), D^{σ}_{G-L} - the Grunwald-Letnikov fractional differential operator, Γ(·) - gamma function, expansion of factorial to real numbers.

From formula (1) the fractial differential can be expressed as follows:

$$\Delta^{\sigma} s(x) = \frac{1}{\Gamma(-\sigma)} \sum_{k=0}^{N-1} \frac{\Gamma(k-\sigma)}{\Gamma(k+1)} * s\left(x - k\left(\frac{x-a}{N}\right)\right), \tag{2}$$

Formula (2) represents the fractial differential for the signal s(x), specifically the numerical approximation of the Grunwald-Letnikov fractial differential.

The Caputo [13] fractional differential can be expressed as:

$$D^{\sigma} f(t) = \frac{1}{\Gamma(n-\sigma)} \int_a^t \frac{f^{(n)}(\tau)}{(t-\tau)^{\sigma-n+1}} d\tau, \tag{3}$$

$D^{\sigma} f(t)$ — the Caputo fractional differential of the function f(t) of order σ, $f^{(n)}(\tau)$ –the classical n-th derivative of f(t), t – the independent variable.

The Atangana-Beleanu derivative [11] can be expressed as:

$${}^{ABC}D^{\sigma}_{a} f(t) = \frac{B(\sigma)}{1-\sigma} \int_a^t f'(\tau) E_{\sigma}\left(-\frac{(t-\tau)^{\sigma}}{1-\sigma}\right) d\tau, \tag{4}$$

${}^{ABC}D^{\sigma}_{a} f(t)$- the Atangana-Baleanu fractional derivative of a function f(t) of order σ, B(·) – the normalization constant, E_{σ} -the generalized gamma function of Mittag-Leffler, f′(τ) – the first derivative of a function f(t), a - the lower limit of integration.

The Mittag-Leffler function can be expressed as:

$$E_{\sigma}(t) = \sum_{k=0}^{\infty} \frac{t^k}{\Gamma(\sigma k + 1)}, \ \sigma > 0, \tag{5}$$

$E_{\sigma}(t)$ - the Mittag-Leffler kernel, which provides smoother behavior of the derivative compared to classical fractional derivatives, and makes it less sensitive to initial conditions.

The normalization function· $\mathcal{M}(\sigma)$·can be expressed as:

$$\mathcal{M}(\sigma) = 1 - \sigma + \frac{\sigma}{\Gamma(\sigma)}, \tag{6}$$

The Atangana-Beleanu fractial integral has the following form:

$$\mathcal{J}^{\sigma}\mathcal{K}(t) = \frac{1-\sigma}{\mathcal{M}(\sigma)}\mathcal{K}(t) + \frac{\sigma}{\Gamma(\sigma)\mathcal{M}(\sigma)} \int_0^t \mathcal{K}(\tau)(t-\tau)^{\sigma-1} d\tau \tag{7}$$

Based on the mathematical framework described in this section, we will apply numerical methods to create approximation polynoms and select approximation coefficients of the Atangana-Beleanu integral.

2.2. Approximation of the Atangana-Baleanu aractial operator

In this section, the following approximations of the Atangana-Baleanu fractial operator are reviewed: approximation based on the Grunwald-Letnikov formula, the Toufiq-Atangana

approximation [13], the approximation based on Euler's method [14], and the approximation based on the midpoint approach. The definition of the fractial integral (7) will serve as the foundation for future algorithms.

2.2.1. The approximation based on the Grunwald-Letnikov formula (GL)

Let's take the Grunwald-Letnikov integral as the basis:

$$\begin{aligned} \mathcal{J}^{\sigma}\ \mathcal{K}(t) &\approx \int_0^t \frac{\mathcal{K}(\tau)}{(t-\tau)^{1-\sigma}} d\tau \\ &= \lim_{h\to 0} h^{-\sigma} \left(\mathcal{K}(t) + \sigma \mathcal{K}(t-h) + \frac{(-\sigma)(-\sigma+1)}{2}\mathcal{K}(t-2h)\right., \\ &\quad + \cdots + \frac{\Gamma(-\sigma+1)}{k!\,\Gamma(-\sigma-N+1)}\mathcal{K}(t-Nh)) \end{aligned} \tag{8}$$

By accepting the approximation step h = 1 for formula (8), and substituting it into formula (7), we obtain the approximation expression:

$$\begin{aligned} \mathcal{J}^{\sigma}_{\mathrm{GL}}\mathcal{K}(t) \approx & \frac{1}{\mathcal{M}(\sigma)}\mathcal{K}(t_n) + \frac{\sigma^2}{\mathcal{M}(\sigma)}\mathcal{K}(t_{n-1}) + \frac{\sigma^3-\sigma^2}{2\mathcal{M}(\sigma)}\mathcal{K}(t_{n-2}) \\ & + \frac{\sigma^2-\sigma^4}{6\mathcal{M}(\sigma)}\mathcal{K}(t_{n-3}) + \ldots \end{aligned} \tag{9}$$

According to formula (9), the coefficients of the polynom are equal to:

$$\begin{aligned} C_0 &= \frac{1}{\mathcal{M}(\sigma)} \\ C_1 &= \frac{\sigma^2}{\mathcal{M}(\sigma)} \\ C_2 &= \frac{\sigma^3-\sigma^2}{2\mathcal{M}(\sigma)} \end{aligned} \tag{10}$$

2.2.2. Approximation based on the Toufic-Atangana approach (T)

This approach involves interpolating the function $\mathcal{K}(\tau)$ over the interval $[t_k, t_{k+1}]$ using the two-step Lagrange interpolation:

$$\mathcal{K}(\tau) = \frac{\mathcal{K}(t_k)}{t_k - t_{k-1}}(\tau - t_{k-1}) + \frac{\mathcal{K}(t_{k-1})}{t_k - t_{k-1}}(\tau - t_k) \tag{11}$$

Substituting $t = t_n$ into formula (7) gives us the following formula:

$$\begin{aligned} \mathcal{J}^{\sigma}_{t}\mathcal{K}(t_n) &= \frac{1-\sigma}{\mathcal{M}(\sigma)}\mathcal{K}(t_n) + \frac{\sigma}{\Gamma(\sigma)\mathcal{M}(\sigma)}\int_0^{t_{n+1}} \mathcal{K}(\tau)(t-\tau)^{\sigma-1}d\tau \\ &= \frac{1-\sigma}{\mathcal{M}(\sigma)}\mathcal{K}(t) + \frac{\sigma}{\Gamma(\sigma)\mathcal{M}(\sigma)}\sum_{k=0}^{n}\int_{t_k}^{t_{k+1}} \mathcal{K}(\tau)(t-\tau)^{\sigma-1}d\tau \end{aligned} \tag{12}$$

Substituting (11) into (12) and performing the transformation with [13], we obtain the following formula:

$$\mathcal{J}_t^{\sigma}\mathcal{K}(t_n) = \frac{1-\sigma}{\mathcal{M}(\sigma)}\mathcal{K}(t_n) + \frac{\sigma h^{\sigma}}{\mathcal{M}(\sigma)\Gamma(\sigma+2)} \times \sum_{m=0}^{n} \left(\mathcal{K}(t_m)\left[(n-m+1)^{\sigma}(n-m+2+\sigma) - (n-m)^{\sigma}(n-m+2+2\sigma)\right] - \mathcal{K}(t_{m-1})\left[(n-m+1)^{\sigma+1} - (n-m)^{\sigma}(n-m+1+\sigma)\right]\right) \tag{13}$$

By expanding the expression under the sum and rewriting the polynom in the following form:

$$\begin{aligned}\mathcal{J}_{\mathrm{TA}}^{\sigma}\mathcal{K}(t_n) = &\left[\frac{(1-\sigma)\Gamma(\sigma+2) + \sigma h^{\sigma}(\sigma+2)}{B(\sigma)\Gamma(\sigma+2)}\right]\mathcal{K}(t_n) \\ &+ \left[\frac{\sigma h^{\sigma}\left((\sigma+3)2^{\sigma} - 2\sigma - 4\right)}{B(\sigma)\Gamma(\sigma+2)}\right]\mathcal{K}(t_{n-1}) \\ &+ \left[\frac{\sigma h^{\sigma}\left((\sigma+4)3^{\sigma} - (2\sigma+5)2^{\sigma} + \sigma + 2\right)}{B(\sigma)\Gamma(\sigma+2)}\right]\mathcal{K}(t_{n-2}) \\ &+ \left[\frac{\sigma h^{\sigma}\left((\sigma+5)4^{\sigma} - (2\sigma+6)3^{\sigma} + (\sigma+3)2^{\sigma}\right)}{B(\sigma)\Gamma(\sigma+2)}\right]\mathcal{K}(t_{n-3}) + \cdots\end{aligned} \tag{14}$$

The coefficients of the polynom (14) are equal to:

$$\begin{aligned}C_0 &= \frac{(1-\sigma)\Gamma(\sigma+2) + \sigma(\sigma+2)}{B(\sigma)\Gamma(\sigma+2)} \\ C_1 &= \frac{(\sigma^2+3\sigma)2^{\sigma} - 2\sigma^2 - 4\sigma}{B(\sigma)\Gamma(\sigma+2)} \\ C_2 &= \frac{(\sigma^2+4\sigma)3^{\sigma} - (2\sigma^2+5\sigma)2^{\sigma} + \sigma^2 + 2\sigma}{B(\sigma)\Gamma(\sigma+2)}\end{aligned} \tag{15}$$

2.2.3. Approximation based on the Euler method (Eu)

According to the Euler method described in [14], we approximate (7) at $t = t_n$ as follows:

$$\mathcal{J}_{\mathrm{Eu}}^{\sigma}\mathcal{K}(t_n) = \frac{1-\sigma}{\mathcal{M}(\sigma)}\mathcal{K}(t_n) + \frac{\sigma h^{\sigma}}{\mathcal{M}(\sigma)\Gamma(\sigma+1)}\sum_{m=0}^{n-1}\theta_{n,m}\mathcal{K}(t_m), \tag{16}$$

where $\theta_{n,m}$:

$$\theta_{n,m} = (n-m)^{\sigma} - (n-m-1)^{\sigma} \tag{17}$$

Substitute expression (17) into (16) and rewrite the polynom without the sign of the sum:

$$\begin{aligned}\mathcal{J}_{\mathrm{Eu}}^{\sigma}\mathcal{K}(t_n) = &\left[\frac{1-\sigma}{B(\sigma)}\right]\mathcal{K}(t_n) + \left[\frac{\sigma h^{\sigma}}{B(\sigma)\Gamma(\sigma+1)}\right]\mathcal{K}(t_{n-1}) + \left[\frac{\sigma h^{\sigma}(2^{\sigma}-1)}{B(\sigma)\Gamma(\sigma+1)}\right]\mathcal{K}(t_{n-2}) \\ &+ \left[\frac{\sigma h^{\sigma}(3^{\sigma}-2^{\sigma})}{B(\sigma)\Gamma(\sigma+1)}\right]\mathcal{K}(t_{n-3}) + \cdots\end{aligned} \tag{18}$$

The coefficients of the polynom (18) are equal to:

$$C_0 = \frac{1-\sigma}{B(\sigma)} \tag{19}$$
$$C_1 = \frac{\sigma}{B(\sigma)\Gamma(\sigma+1)}.$$
$$C_2 = \frac{\sigma\,(2^\sigma - 1)}{B(\sigma)\Gamma(\sigma+1)}$$

2.2.4. Approximation based on midpoint approach (MP)

Let's rewrite formula (7) by replacing $\tau = t - \omega$ in the integral, we have:

$$\mathcal{J}^\sigma \mathcal{K}(t) = \frac{1-\sigma}{\mathcal{M}(\sigma)}\mathcal{K}(t) + \frac{\sigma}{\Gamma(\sigma)\mathcal{M}(\sigma)}\int_0^t \frac{\mathcal{K}(t-\tau)}{\tau^{1-\sigma}}d\tau \tag{20}$$

By dividing the integral in (20), we obtain:

$$\mathcal{J}_t^\sigma \mathcal{K}(t) = \frac{1-\sigma}{\mathcal{M}(\sigma)}\mathcal{K}(t) + \frac{\sigma}{\Gamma(\sigma)\mathcal{M}(\sigma)}\sum_{m=0}^{n-1}\int_{t_m}^{t_{m+1}} \frac{\mathcal{K}(t-\tau)}{\tau^{1-\sigma}}d\tau \tag{21}$$

The integral in formula (21) can be approximated as:

$$\int_{t_m}^{t_{m+1}} \frac{\mathcal{K}(\tau)}{\tau^{1-\sigma}}d\tau \approx \frac{\mathcal{K}(t_m) + \mathcal{K}(t_{m+1})}{2}\int_{t_m}^{t_{m+1}} \frac{d\tau}{\tau^{1-\sigma}} \tag{22}$$

Substituting formula (21) into (22) and taking $t = t_n = nh$, we obtain:

$$\begin{aligned}\mathcal{J}^\sigma \mathcal{K}(t_n) =& \frac{1-\sigma}{\mathcal{M}(\sigma)}\mathcal{K}(t_n) \\ &+ \frac{\sigma}{\Gamma(\sigma)\mathcal{M}(\sigma)}\sum_{m=0}^{n-1}\frac{\mathcal{K}(t_n - t_m) + \mathcal{K}(t_n - t_{m+1})}{2\sigma}[((m+1)h)^\sigma - (mh)^\sigma \\ =& \frac{1-\sigma}{\mathcal{M}(\sigma)}\mathcal{K}(t_n) + \frac{h^\sigma}{\Gamma(\sigma)\mathcal{M}(\sigma)}\sum_{m=0}^{n-1}\frac{\mathcal{K}(t_{n-m}) + \mathcal{K}(t_{n-m-1})}{2}[(m+1)^\sigma - m\end{aligned} \tag{23}$$

By performing arithmetic calculations, formula (23) can be reduced to the following form:

$$\begin{aligned}\mathcal{J}_{\mathrm{MP}}^\sigma \mathcal{K}(t_n) =& \left[\frac{1-\sigma}{\mathcal{M}(\sigma)} + \frac{h^\sigma}{2\mathcal{M}(\sigma)\Gamma(\sigma)}\right]\mathcal{K}(t_n) \\ &+ \left[\frac{h^\sigma}{2B(\sigma)\Gamma(\sigma)}(2^\sigma)\right]\mathcal{K}(t_{n-1}) + \left[\frac{h^\sigma}{B(\sigma)\Gamma(\sigma)}\left(\frac{3^\sigma - 1}{2}\right)\right]\mathcal{K}(t_{n-2}) \\ &+ \left[\frac{h^\sigma}{B(\sigma)\Gamma(\sigma)}\left(\frac{4^\sigma - 2^\sigma}{2}\right)\right]\mathcal{K}(t_{n-3}) + \cdots\end{aligned} \tag{24}$$

The coefficients of the polynom (24) are equal to:

$$C_0 = \frac{2\Gamma(\sigma)(1-\sigma)+1}{2\mathcal{M}(\sigma)\Gamma(\sigma)}$$
$$C_1 = \frac{2^{\sigma-1}}{\mathcal{M}(\sigma)\Gamma(\sigma)} \quad . \tag{25}$$
$$C_2 = \frac{3^{\sigma}-1}{2\mathcal{M}(\sigma)\Gamma(\sigma)}$$

2.3. Fractional masks

Based on the research results [8, 9], for image enhancement in the context of fractal operators, formula (26) is the most important:

$$\mathcal{T}^{\alpha} f(t) \approx S_0 f(t) + \mathrm{S}_1 f(t-1) + \mathrm{S}_2 f(t-2) + \ldots, \tag{26}$$

where, S_0, S_1 ... the sequential coefficients in expansion (26), $f(t)$ –signal value in particular point.

This formula can be extended to the two-dimensional space as follows:

$$^{x}\mathcal{T}\, x^{\alpha} f(x,y) \approx S_0 f(x,y) + \mathrm{S}_1 f(x-1,\mathrm{y}) + \mathrm{S}_2 f(x-2,\mathrm{y}) + \ldots, \tag{27}$$

$$^{y}\mathcal{T}^{\alpha} f(x,y) \approx S_0 f(x,y) + \mathrm{S}_1 f(x,y-1) + \mathrm{S}_2 f(x,y-2) + \ldots, \tag{28}$$

In article [11], masks for edge detection in images were analyzed. In this article, we will focus on the use of masks for noise reduction, with a detailed description provided in [8]. Due to the specific arrangement of coefficients, this mask contribute to noise reduction rather than edge detection.

Figure 1 show mask corresponding to eight symmetric directions. The masks allow calculating the fractional differential in eight directions. The presence of eight masks creates anti-rotation properties, enabling the same processing result regardless of the texture's position in the image. S_0, S_1 ... are the sequential coefficients from formulas (27) and (28).

C_2	0	C_2	0	C_2
0	C_1	C_1	C_1	0
C_2	C_1	$8C_0$	C_1	C_2
0	C_1	C_1	C_1	0
C_2	0	C_2	0	C_2

Figure 1: Denoise mask for eight directions.

2.4. Algorithm and software implamentation

The application of matrices with approximation coefficients in the algorithm for image denoising is described next.

The algorithm consists of the following steps:

1. The original image is read in grayscale. The input image can be of any size and format. The color value is encoded in 1 byte, corresponding to 256 grayscale values.

2. Eight filters W1-W8 (matrices) are created to determine gradients in different directions (0°, 45°, 90°, 135°, ...). Each filter highlights the edges or gradients of the image in a specific direction.
3. For each filter, a convolution operation is performed on the gradient-transformed grayscale image. This allow to extract different edge components of the image according to the filter directions.
4. The results of the convolutions from each filter are combined to create the final image, which represents the aggregate of the extracted gradients in various directions. The combination of gradients is achieved by calculating the arithmetic mean of the pixel values from the eight gradients.
5. The next step is to normalize the pixel values, as the color values may exceed or fall below the range [0; 255].

The algorithm was implemented in Python [15], using libraries for image processing such as OpenCV [16], also libraries NumPy and SciPy [17] were used.

Input data is a very important component of the research. Medical data is often hard to access because it contains confidential information. However, anonymized MRI images are available on certain platforms and can be used for scientific purposes. The dataset from the resource [18] contains hundreds of small-sized MRI images of the brain in .jpg format.

3. Analysis of results

3.1. Visual comparison

The first step in evaluating the algorithm is a visual comparison of the original image and the denoised images obtained using fractial approximations.

We will apply the approximation coefficients in the algorithm described above, using masks of size 5x5 and values in the range of [0.1; 0.9].

3.1.1. Results of the algorithm based on the Grunwald-Letnikov approximation (GL)

In Figure 2 and Figure 3, the original image, noised and the denoise images using the Grunwald-Letnikov approximation are shown. It is noticeable that with an increase in the value of (σ), the noise become less visible. The best result achieved when σ = 0.9 for this approximation.

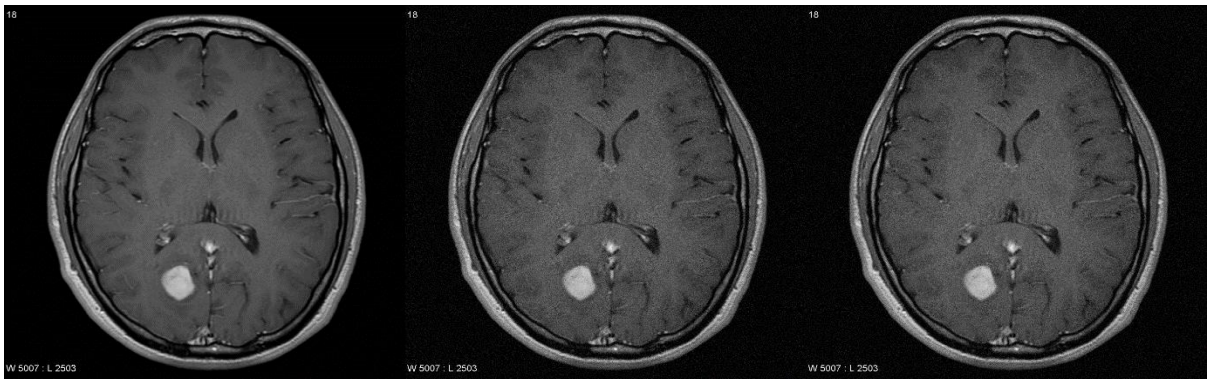


Figure 2: GL Approximation. Original image, noisy image, denoised brain MRI images for σ =0.3 (from left to right).

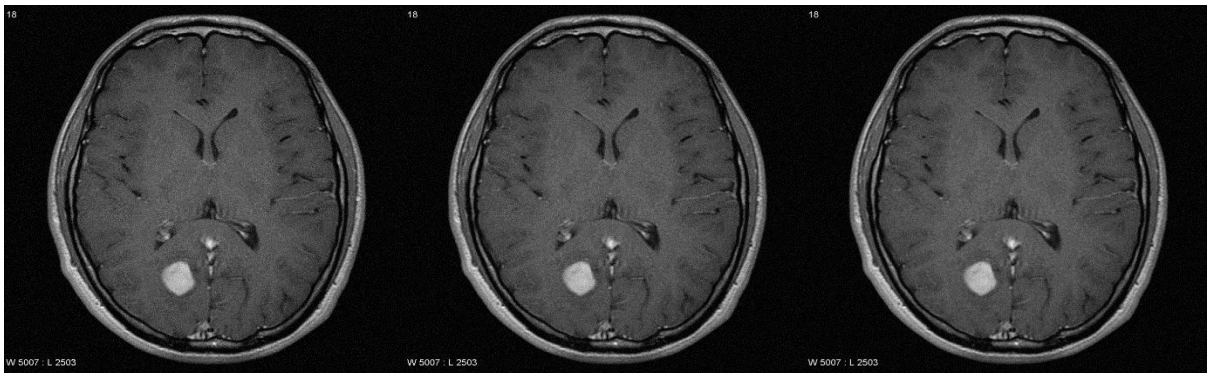


Figure 3: Denoised brain MRI images for σ = {0.5, 0.7, 0.9} (from left to right), GL approximation.

3.1.2. Result of the algorithm's operation for the approximation based on the Toufique-Atangana approach (T)

The original image, noised and denoised image for σ = 0.3 are shown, while Figure 5 displays enhanced images for σ = {0.5, 0.7, 0.9} using the Taufik-Atangan approximation. It is noticeable that there is a progressive improvement in the image denoising, and for σ = 0.9 (Figure 4).

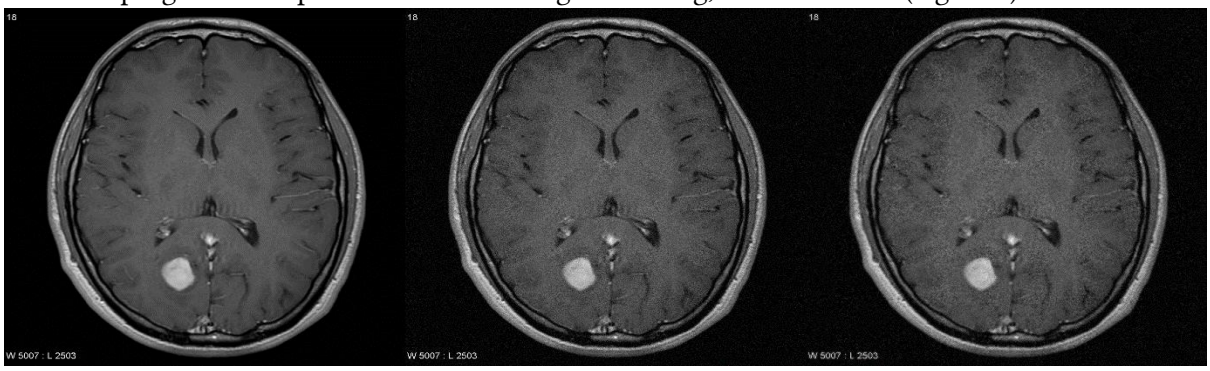


Figure 4: Approximation A. Original image, noisy image, denoised brain MRI images for σ =0.3 (from left to right).

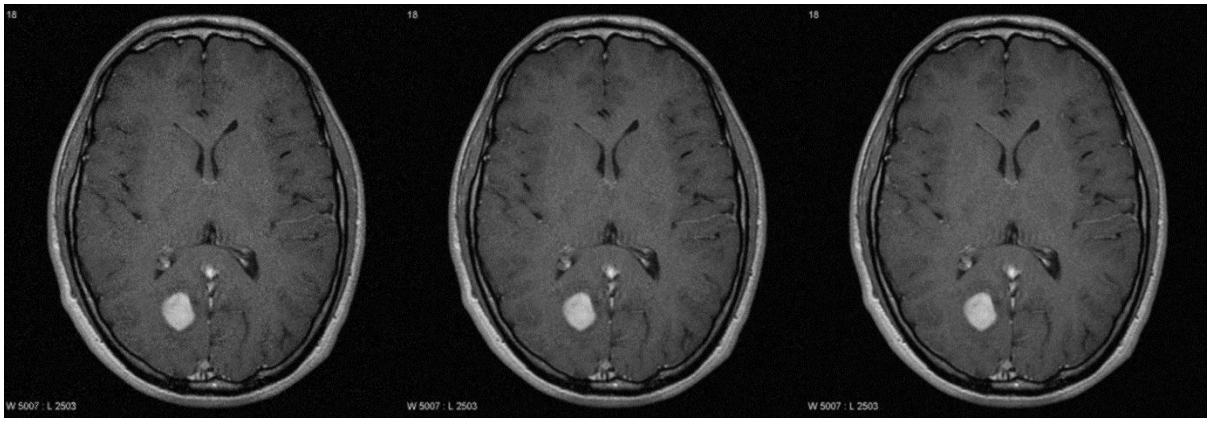


Figure 5: Denoised brain MRI images for σ = {0.5, 0.7, 0.9} (from left to right), approximation A.

3.1.3. Result of the algorithm for approximation based on Euler's method (Eu)

On Figure 6 and Figure 7, the original image, noised and approximations based on the Euler method are shown. This approximation denoise the noised image as the value of σ increases, visually σ=0.9 works best.

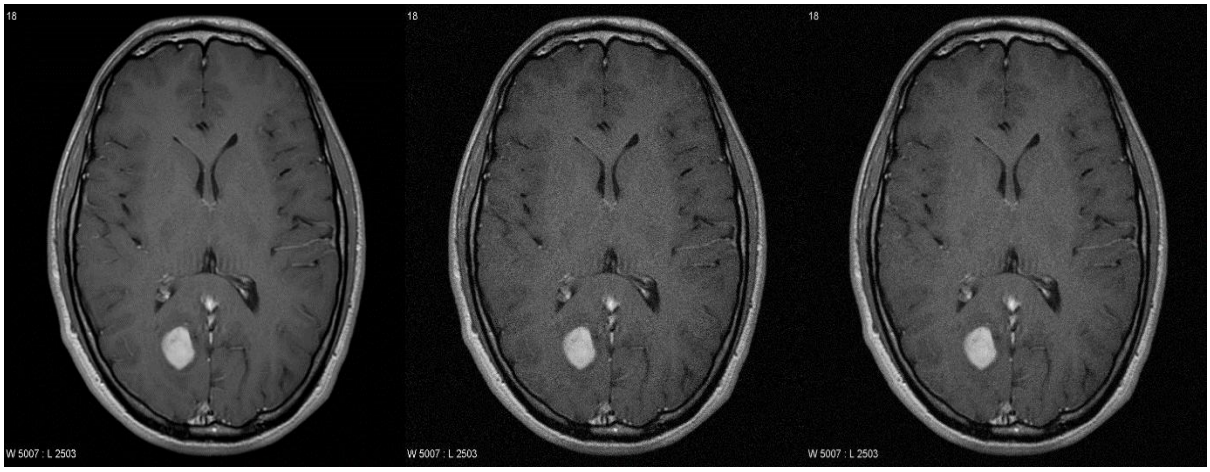


Figure 6: Approximation Eu. Original image, noisy image, denoised brain MRI images for σ =0.3 (from left to right).

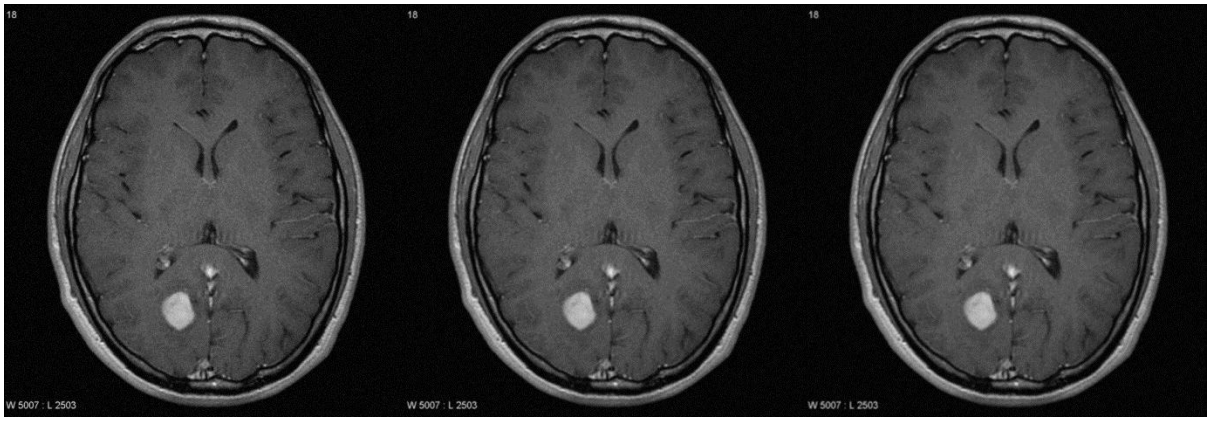


Figure 7: Denoised brain MRI images for σ = {0.5, 0.7, 0.9} (from left to right), approximation Eu.

3.1.4. Result of the algorithm's operation for approximation based on the midpoint method. (MP)

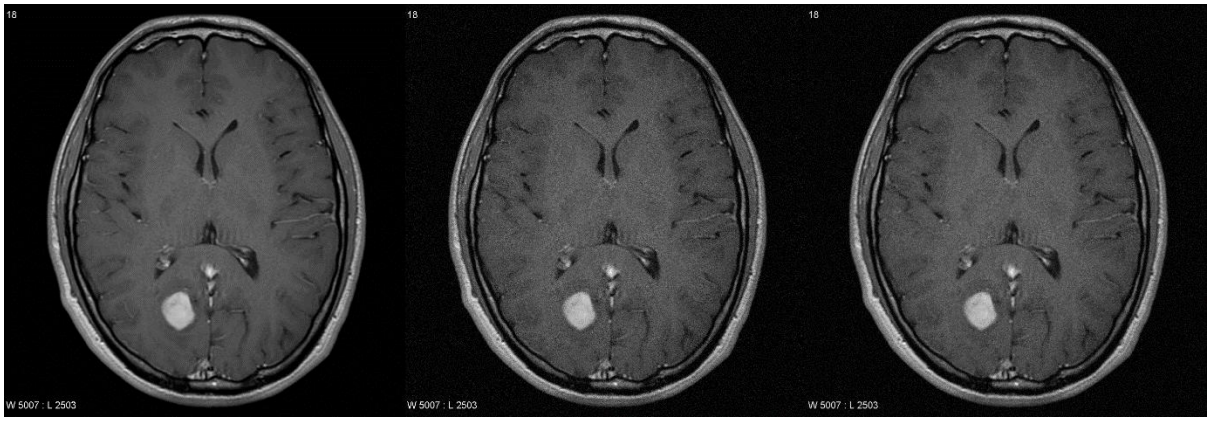


Figure 8: Approximation Mp. Original image, noisy image, denoised brain MRI images for σ =0.3 (from left to right).

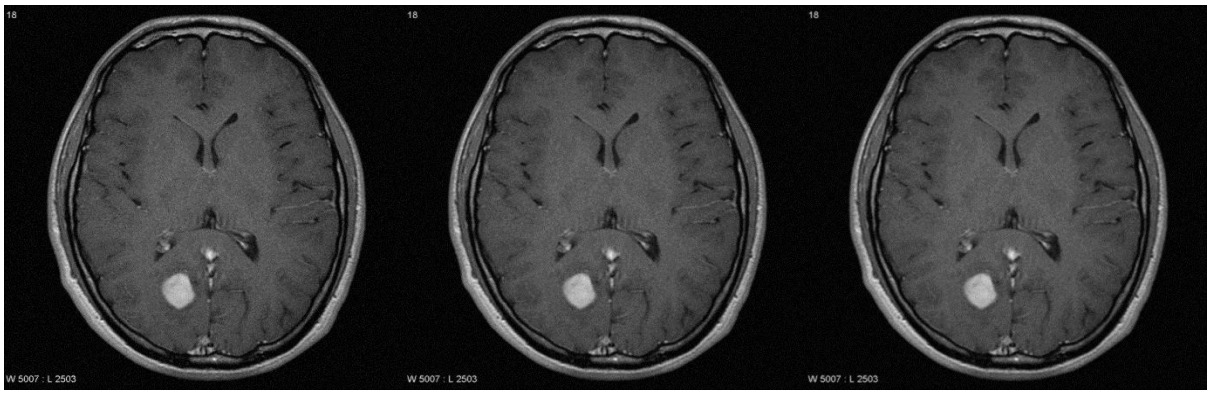


Figure 9: Denoised brain MRI images for σ = {0.5, 0.7, 0.9} (from left to right), approximation Mp.

This approximation also shows gradual image denoising as the parameter σ increases.

3.2. Comparation of image parameters

To evaluate the quality of an image after applying enhancement techniques like denoising, it's important to look at two key measures: PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index).

PSNR helps quantify how much the processed image differs from the original, especially after noise reduction. A higher PSNR indicates fewer distortions, meaning the processed image retains better quality. Essentially, it tells you how close the filtered image is to the original in terms of pixel values.

On the other hand,SSIM goes beyond just pixel comparison and looks at the image's structure, contrast, and brightness. It's more in line with how the human eye perceives image quality. SSIM values range from -1 to 1, with values closer to 1 indicating that the two images are highly similar in structure.

In summary, using both PSNR and SSIM together gives a more complete picture of image quality: PSNR provides a straightforward numerical measure of any loss, while SSIM offers insight into how similar the images look to the human eye.

PSNR can be calculated as:

$$PSNR = 10 \log_{10}\left(\frac{MAX^2}{MSE}\right), \tag{29}$$

where MAX, is the maximum possible pixel value in the image. For example, in an 8-bit image, where pixel intensity ranges from 0 to 255, MAX equals 255. MSE (Mean Squared Error) is the mean squared difference between the original and processed images. It is calculated as:

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [I(i,j) - K(i,j)]^2, \tag{30}$$

where, I(i,j) is the pixel value at position (i,j) in the original image, K(i,j) is the pixel value at position (i,j) in the processed image, M and N are the number of rows and columns in the image, respectively.

$$SSIM = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}. \tag{31}$$

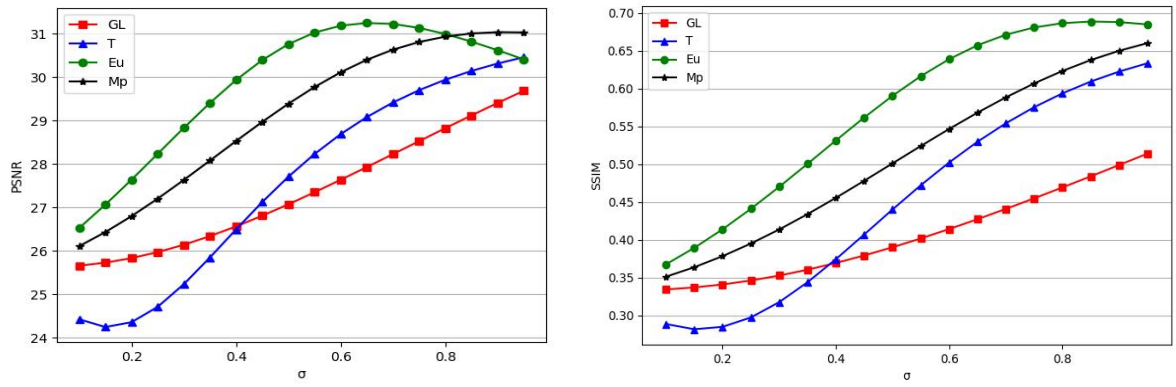
where, x and y are two images, μ_x and μ_y are the mean values (luminance) of x and y, σ_x^2 and σ_y^2 are the variances (contrast) of the x and y, σ_{xy} is the covariance between the x and y, C1and C2 are constants to avoid instability when the denominators are close to zero.

From the graphs in Figure 10 all four approximation methods demonstrate an increasing trend in PSNR and SSIM values with higher σ values. The only exception is the method based on T, which shows a decline for small σ values.

The GL-based algorithm showed the worst image improvement among the observed parameters PSNR and SSIM. The PSNR starts at 26 and reaches around 29, indicating lower efficiency compared to other methods. SSIM values fluctuate between 0.38 and 0.5, indicating lower structural similarity of images compared to the other methods.

The T-based approximation algorithm shows better results than GL in the interval from 0.4. For lower values, the PSNR and SSIM values are the worst among the compared methods.

For the Euler (Eu) algorithm, PSNR values range from 26 to 31, which also indicates good quality, although lower than that of Eu. SSIM values increase from 0.4 to approximately 0.6. Although this method is less effective than Eu, it demonstrates satisfactory image quality while maintaining structure.



a) b)

Figure 10: (a) PSNR values, (a) SSIM values for different values of σ, for approximations: a) GL, b) T, c) Eu, d) MP

The MP-based algorithm starts with a PSNR of approximately 26 at σ=0 and reaches a maximum of about 32 at σ=0.6. This indicates high effectiveness in preserving image quality during processing. SSIM values start at 0.4 and rise to 0.65 at σ=0.8. The high SSIM value confirms that this method accurately reflects the structural similarity between the original and processed images.

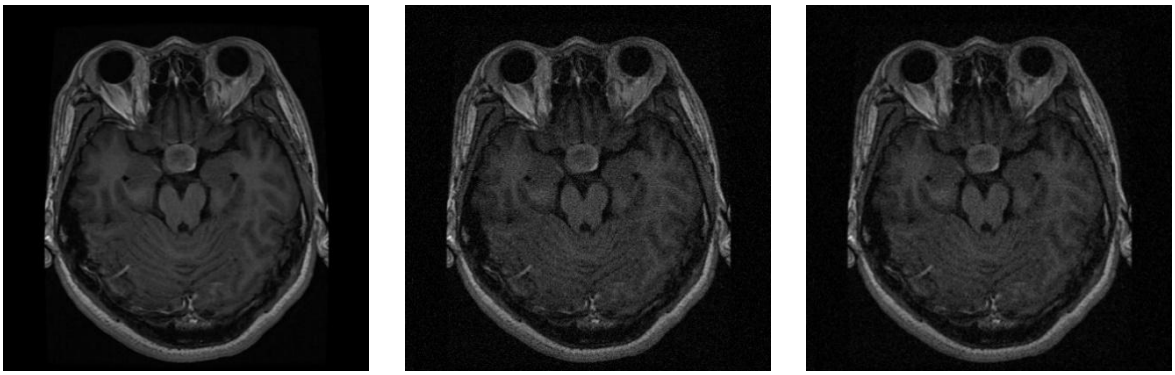
In summary, based on the observations above, it can be stated that the best results are demonstrated by the Euler (Eu) approximation method in the range of σ=0.7 - 0.8.

3.3. Comparison of the result with other algorithms

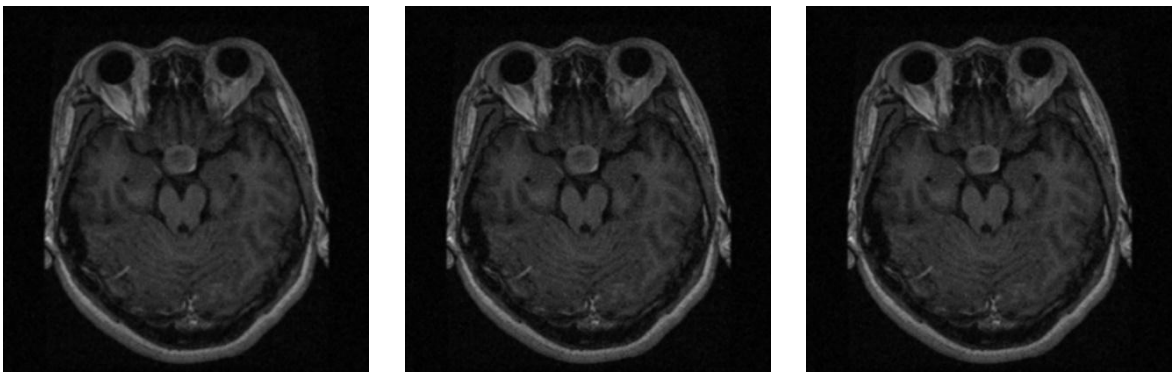
We will conduct a visual and later an analytical comparison of the results of approximation algorithms with other image denoising algorithms. First of all it is algorithm YiFePu3 [9], which reflects the authors' surnames in the article and based on Riemann-Liouville integral approximation. Another algorithm which is widely used in the field of computer vision, is Median based filter [2].

Figure 11 shows the original image (a), noised original imgae (b), images denoised using the Atangana-Baleanu fractial operator approximations (c), (d), (e), (f), the YiFePu3 fractial operator (g), as well as those denoised by the Median filter algorithm (h).

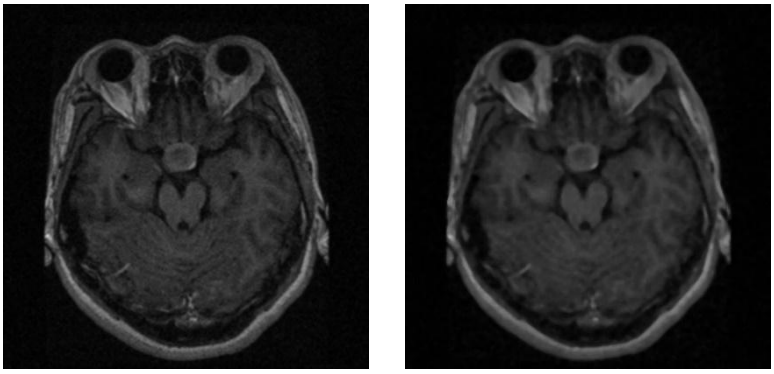
Visually, the best results were achieved by the Atangana-Baleanu approximations (d),(e),(f) and YiFePu3 (g) algorithm. It is noticeable reduction in noise visibility and its smoothing in the denoised images. In addition, there is a preservation of texture in the image, as well as details in relatively smooth areas of the denoised images. For image (c) the noise is much more visible, but still enhancement is present. The median filter algorithm (h) has several drawbacks, one of which is that it can cause slight blurriness in images, particularly affecting fine details. As a result, both small details and the edges of textural elements appear blurred in the image.



a) b) c)



d) e) f)



g) h)

Figure 11: MRI of the brain: (a) original image, (b) noised image, (c) GL algorithm (σ =0.9), (d) T algorithm (σ =0.9), (e) Eu algorithm (σ =0.7), (f) YiFePu3 algorithm (σ =0.9), (g) Median filter

Next, we will consider the comparison of PSNR and SIMM for different algorithms.

Table 1
The PSNR values of the enhanced images

σ value	GL	A	Eu	MP
0.1	25.8068	24.5803	26.6835	26.2605
0.2	25.9824	24.5187	27.8243	26.9613
0.3	26.2923	25.4018	29.0878	27.82
0.4	26.7207	26.6971	30.3261	28.7658
0.5	27.2401	27.9798	31.3511	29.7069
0.6	27.8217	29.0606	32.0007	30.5472
0.7	28.441	29.9044	**32.2276**	31.209
0.8	29.0787	30.5416	32.1139	31.653
0.9	**29.7184**	**31.0179**	31.7932	**31.8845**

Table 2
The SSIM values of the enhanced images

σ value	GL	A	Eu	MP
0.1	0.3716	0.3199	0.4055	0.3889
0.2	0.3785	0.3107	0.4514	0.4162
0.3	0.3907	0.3389	0.5041	0.4506

0.4	0.4078	0.3912	0.5579	0.4899
0.5	0.4288	0.4514	0.6053	0.5304
0.6	0.4526	0.507	0.6401	0.5685
0.7	0.4784	0.5521	0.6593	0.601
0.8	0.5051	0.5861	**0.664**	0.6259
0.9	**0.5322**	**0.6108**	0.6577	**0.6427**

Table 3
The values of PSNR and SSIM for denoised image with fractional integaral YiFePu3(σ 0.9) and Median filter

Image	PSNR	SSIM
YiFePu3(σ = 0.5)	**31.9774**	0.6504
Median filter	31.2501	**0.7447**

According to Table 3, it can be stated that the highest values of PSNR is for the YiFePu3 algorithm, which aligns with the visual comparisons in Figure 11 (g). At the same time the hightest values of SSIM is for Median filter (h), hight SSIM indicates only similarity in texture but does not guarantee preservation of details.

For the algorithms based on the approximations of the Atangana-Baleanu integral, Table 1 shows that the maximum value of PSNR is achieved at the highest values of σ, except for the algorithm based on Eu approximation. For this algorithm, the maximum value of PSNR occurs at σ = 0.7, as for larger values of v PSNR do not become bigger. The SSIM values form Table 2 show almost the same behavior except that best SSIM for Eu method is for σ = 0.8 .

Thus, to summarize that in comparison, the YiFePu3 approximation (GL) and (Eu) Atangana-Baleanu show almost comparable results, while the approximations (A) and (Mp) show worst results, and GL the worst result. At the same time, algorithms based on fractial operators show significantly better results compared to standard algorithm of Median filter.

4. Conclusion

The algorithm based on the Atangana-Baleanu fractial operator can be effectively used to denoise image and improve the quality of MRI images, as confirmed by the results of the visual comparisons conducted. The analysis also showed that the enhanced images demonstrate hight values of PNSR and SSIM. High PSNR values indicate less distortion in the image and its proximity to the original, suggesting effective noise reduction. High SIMM values indicate a high structural similarity between the denoised and original images. Thus, the results of the visual comparison and numerical indicators confirm the effectiveness of the algorithm in denoising images, improving their quality and informativeness. Algorithm GL performed the worst for noise removal, while algorithm A showed better results. Algorithm Mp further improved the performance, but algorithm Eu achieved the best noise reduction outcomes.Each algorithm can be used with params σ = [0.1, 0.9].

When compared to other algorithms, it is evident that algorithms based on fractial operators (Atangana-Baleanu and YiFePu3 - Riemann-Liouville) show better results in image denoising

compared to standard image denoising methods. Moreover, the algorithm based on (Eu) demonstrate better results than YiFePu3.

This study confirmed the properties of fractial operators to remove hight-frequency noise elements of images. Therefore, fractial operators are advisable for denoising of medical images, and further research in this direction is relevant.

5. Declaration on Generative AI

During the preparation of this paper, the authors utilized Grammarly to verify spelling and grammar accuracy. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

6. References

[1] D'Haeyer, P.F. Johannes, Gaussian filtering of images: A regularization approach. Signal Processing, volume 18, issue 2, 1989, pp. 169–181. doi:10.1016/0165-1684(89)90048-0.

[2] H.-L. Eng, K.-K. Ma, Noise adaptive soft-switching median filter, IEEE Transactions on Image Processing, vol. 10, 2001, pp. 242–251.

[3] W.K. Pratt, Digital Image Processing: PIKS Inside, Wiley, New York, 2001.

[4] Y. Sokolovskyy, M. Levkovych, O. Mokrytska, V. Atamanvuk, Mathematical Modeling of Two-Dimensional Deformation-Relaxation Processes in Environments with Fractal Structure, in: Proceedings of the 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP), Lviv, Ukraine, 2018, pp. 375–380. doi:10.1109/DSMP.2018.8478569.

[5] Y. Sokolovskyy, M. Levkovych, O. Mokrytska, Y. Kaspryshyn, Mathematical Modeling of Nonequilibrium Physical Processes, Taking into Account the Memory Effects and Spatial Correlation, in: Proceedings of the 2019 9th International Conference on Advanced Computer Information Technologies (ACIT), Ceske Budejovice, Czech Republic, 2019, pp. 56–59. doi:10.1109/ACITT.2019.8780011.

[6] Y. Sokolovskyy, M. Levkovych, O. Mokrytska, Y. Kaplunskyy, Mathematical models of biophysical processes taking into account memory effects and self-similarity, CEUR Workshop Proceedings, vol. 2255, pp. 215–228, 2018.

[7] Y. Sokolovskyy, D. Manokhin, O. Mokrytska, Segmentation of Medical Images Using Deep Learning and Texture Enhancement Based on Fractional Derivative Operators, in: Proceedings of the 2024 IEEE 19th International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH), Zozuli, Ukraine, 2024, pp. 113–118.

[8] Q. Yang, D. Chen, T. Zhao, et al., Fractional Calculus in Image Processing: A Review, FCAA, vol. 19, 2016, pp. 1222–1249. doi:10.1515/fca-2016-0063.

[9] Y.-F. Pu, J.-L. Zhou, X. Yuan, Fractional Differential Mask: A Fractional Differential-Based Approach for Multiscale Texture Enhancement, IEEE Transactions on Image Processing, vol. 19, no. 2, Feb. 2010, pp. 491–511. doi:10.1109/TIP.2009.2035980.

[10] A. Atangana, New fractional derivatives with nonlocal and non-singular kernel: Theory and application to heat transfer model, Thermal Science, vol. 20, 2016. doi:10.2298/TSCI160111018A.

[11] B. Ghanbari, A. Atangana, Some new edge detecting techniques based on fractional derivatives with non-local and non-singular kernels, Advances in Difference Equations, 2020. doi:10.1186/s13662-020-02890-9.

[12] I. Podlubny, Fractional Differential Equations, Mathematics in Science and Engineering, vol. 198, 1999.

[13] M. Toufik, A. Atangana, New numerical approximation of fractional derivative with non-local and non-singular kernel: Application to chaotic models, European Physical Journal, vol. 132, no. 10, p. 444, 2017.

[14] C. Li, F. Zeng, The finite difference methods for fractional ordinary differential equations, Numerical Functional Analysis and Optimization, vol. 34, no. 2, 2013, pp. 149–179.
[15] G. Bradski, The OpenCV Library, Dr. Dobb's Journal of Software Tools, 2000.
[16] G. Van Rossum, F.L. Drake Jr., Python Reference Manual, PythonLabs, 2001.
[17] P. Virtanen, R. Gommers, T.E. Oliphant, et al., SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, Nature Methods, vol. 17, 2020, pp. 261–272.
[18] Kaggle, Brain MRI images for brain tumor detection, 2020. URL: https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection/data.
[19] R.M. Haralick, K. Shanmugam, I. Dinstein, Textural Features for Image Classification, IEEE Transactions on Systems, Man, and Cybernetics, vol. 3, no. 6, 1973, pp. 610–621.
[20] H. Greenspan, C.H. Anderson, S. Akber, Image enhancement by nonlinear extrapolation in frequency space, IEEE Transactions on Image Processing, vol. 9, no. 6, Jun. 2000, pp. 1035–1048.
[21] S. Dippel, M. Stahl, R. Wiemker, T. Blaffert, Multiscale contrast enhancement for radiographies: Laplacian pyramid versus fast wavelet transform, IEEE Transactions on Medical Imaging, vol. 21, no. 4, 2002, pp. 343–353.
[22] C.E. Shannon, A Mathematical Theory of Communication, Bell System Technical Journal, vol. 27, no. 3, 1948, pp. 379–423.