

# Radiographic Images Fractional Edge Detection based on Genetic Algorithm

WESSAM S. ELARABY<sup>\*</sup>, AHMED H. MADIAN<sup>\*,\*\*\*</sup>, MAHMOUD A. ASHOUR<sup>\*</sup>,  
IBRAHIM FARAG<sup>\*\*</sup>, MOHAMMAD NASSEF<sup>\*\*</sup>

<sup>\*</sup>Radiation Engineering Department, NCRRT, Egyptian Atomic Energy Authority, CAIRO, EGYPT

<sup>\*\*</sup>Department of Computer Science, Faculty of computing and Information, Cairo University, CAIRO, EGYPT

<sup>\*\*\*</sup>Nanoelectronics Integrated System Center (NISC), Nile University, CAIRO, EGYPT  
Email: [eng.wessamsayed@yahoo.com](mailto:eng.wessamsayed@yahoo.com), [amadian@nu.edu.eg](mailto:amadian@nu.edu.eg), [ma\\_ashour53@eaea.org.eg](mailto:ma_ashour53@eaea.org.eg),  
[i.farag@fci-cu.edu.eg](mailto:i.farag@fci-cu.edu.eg), [m.nassef@fci-cu.edu.eg](mailto:m.nassef@fci-cu.edu.eg)

*Abstract:* - Recently, fractional edge detection algorithms have gained focus of many researchers. Most of them concern on the fractional masks implementation without optimization of threshold levels of the algorithm for each image. One of the main problems of the edge detection techniques is the choice of optimal threshold for each image. In this paper, the genetic algorithm has been used to enhance the selection of the threshold levels of the edge detection techniques for each image automatically. A fully automatic way to cluster an image using K-means principle has been applied to different fractional edge detection algorithms to extract required number of thresholds. A performance comparison has been done between different fractional algorithms with and without genetic algorithm. Evaluation of the noise performance upon the addition of salt and pepper noise is measured through the peak signal to noise ratio (PSNR) and bit error rate (BER) simulated by using MATLAB.

*Key-Words:* - Edge Detection, Fractional Systems, Soft Computing Techniques, Biomedical, Genetic Algorithm, clustering-Kmean

## 1 Introduction

Medical imaging has gained focus of many researchers as it played a very important role in the study and early diagnosis of a lot of diseases over the past five decades [1]. The medical images are mostly used as radiographic techniques to help in early diagnosis, curing and studies [2].

Nowadays, digital image processing is ameliorated by using the image enhancement techniques for additional processing [3]. Image segmentation has the goal to extract the information which is the first step in image analysis [4]. It is the method of partitioning the image to extract interest parts in a simple and easy analyzed way [5] [6].

Edge detection can be deemed as one of the most common techniques in many applications in the area of image processing such as biomedical, radiographic images. It has the goal to distinguish and locate the sharp changes in brightness of an image [7] [8].

Edge detection uses the integer-order differential methods. It could enhance the edge

information effectively; however, it could be sensitive to noise and easy to lose image detail information. The fractional-order derivative has been applied to the edge detection methods to solve this problem [9].

It is still a major challenge in image processing to get the optimal threshold for each image, as these traditional techniques have limitations of using the fixed value of thresholds [10]. Soft computing as compared to the traditional techniques, it can deal with the mystery and uncertainty in image processing in a better way. It can build a machine which can work like a human to develop intelligence [11].

In this paper, the main objective is to adapt the selection of thresholds for each image based on genetic algorithm. A fully automatic way to cluster an image using K-means principle has been applied to different fractional edge detection algorithms to extract required number of thresholds that used by the genetic algorithm. Evaluation of the noise performance upon the addition of salt and pepper

noise is measured through the peak signal to noise ratio (PSNR) and bit error rate (BER) simulated by using MATLAB.

## 2 Material and Methods

Recently, lots of research papers concern on fractional calculus [12] that played a vital role in many fields, such as mechanics, robotics and image processing. The development of Fractional calculus has been taken from different views, and the most widely used definitions are the Riemann-Liouville (R-L), Grünwald-Letnikov (G-L), and Caputo fractional differentiation [13]. In [14], a new mask based on the Newton Interpolation’s Fractional Differentiation (NIFD) has been proposed and applied to image edge detection. According to a noisy image, the performance metrics showed that the proposed method gives a better edge information image than sobel and canny operator.

Table 1 discusses the characteristics of fractional order vs integral order differential [15]. From the characteristics, the fractional differential could nonlinearly boost high-frequency marginal information, preserve nonlinearly the low-frequency contour feature and boost nonlinearly texture details. When the image is processed, it needs to keep the original information, improve image quality, boost details and texture characteristics, and keep the marginal details and energy as well. All these requirements are easy to be obtained by the fractional differential-based algorithm.

Table 1: Fractional-order characteristics

	<b>Fractional Differentiation</b>	<b>Integral Differentiation</b>
<b>Smooth area</b>	Non-zero	zero
<b>Initial point of gray scale gradient</b>	Non-zero	zero
<b>Slope</b>	Non-zero or constant	Constant

Nowadays, optimization methods are being used vastly in many sides of image analysis. Optimization can be defined as the process of obtaining the “best” solution to a problem [16]. Evolutionary algorithms are one of the popular stochastic optimization methods that can be used to get nearly optimal solutions to global optimization problems [17]. Evolutionary computation methods evaluate multiple solutions in parallel as opposed to a single candidate solution because of this they are good for global optimization techniques and are less

likely to become stuck in a local optimum. It has many methods such as Genetic Algorithms, Genetic Programming, and Evolutionary Programming [16].

A fully automatic way to cluster an image using K-means principle is one of the clustering algorithm that has been applied to each image without the need to assign the number of clusters. It automatically gets a number of clusters and cluster center iteratively [18]. K-means is considered as one of the simplest and easiest unsupervised learning algorithms. K-means clustering algorithm iteratively computes a mean intensity for each class to cluster the data and classifies each pixel in the class with the closest mean to segment the image. The main goal is to define k centroids, one for each cluster, by taking each point belonging to a given data set and associating it to the nearest centroid until there is no point anymore. After that, continued in a loop until k centroids approach to the main point and a new binding has to be done between the same data set points and the nearest new centroid. K-means is easy, simple and has relatively low computational complexity by comparing to the Fuzzy C-means [18]. The flowchart for the implemented k-means clustering is shown in Fig. 1.

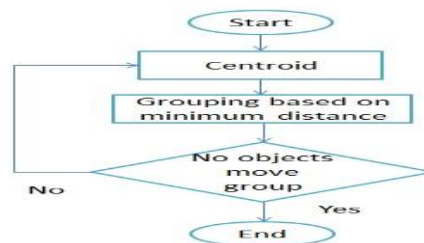


Figure 1: Implemented K-means Clustering Process

In this section, experiments are done on different types of Images using different fractional algorithms. The different algorithms used in this paper using fractional edge detection are shown in table 2.

Table 2: Eight algorithms using the fractional edge detection

<b>Algorithms</b>	<b>Techniques used</b>
Algorithm 1	Fractional_Sobel
Algorithm 2	Fractional_Mask1
Algorithm 3	Fractional_Mask2
Algorithm 4	Fractional_Mask3
Algorithm 5	GA Fractional_Sobel
Algorithm 6	GA Fractional_Mask1
Algorithm 7	GA Fractional_Mask2
Algorithm 8	GA Fractional_Mask3

The first four algorithms use different fractional masks. Apply each fractional mask algorithm with different fractional orders with edge detection. The first step is to read the input image, convolve the image with Gaussian filter, convolute the image by the chosen filter and then apply the fractional algorithm to get the output image. Table 3 shows the different fractional masks and the equations used for each one.

Algorithm 1 uses fractional-order sobel mask. Sobel detection is a 1<sup>st</sup> order derivative edge detection method. It works by detecting edges along the vertical and horizontal axis individually based on a pair of 3×3 convolution mask [19] [20]. The differential form of the gradient components can be found along the x- and y-directions.

$$O_x = \frac{1}{2} \left( \frac{\partial s(x+1,y-1)}{\partial x} + 2 \frac{\partial s(x+1,y)}{\partial x} + \frac{\partial s(x+1,y+1)}{\partial x} \right) \quad (1)$$

$$O_y = \frac{1}{2} \left( \frac{\partial s(x-1,y+1)}{\partial y} + 2 \frac{\partial s(x,y+1)}{\partial y} + \frac{\partial s(x+1,y+1)}{\partial y} \right) \quad (2)$$

The Grünwald-Letnikov definition is used by assuming the size of image  $s$  is  $M \times N$ , and then the discrete form of  $\nabla^v s$  can be represented as [20]

$$(\nabla^v s)_{i,j} = ((\Delta_1^v s)_{i,j}, (\Delta_2^v s)_{i,j}) \quad 1 \leq i \leq M, 1 \leq j \leq N \quad (3)$$

Where

$$\begin{cases} (\Delta_1^v s)_{i,j} = \sum_{n=0}^{n-1} (-1)^n C_n^v s_{i-n,j} \\ (\Delta_2^v s)_{i,j} = \sum_{n=0}^{n-1} (-1)^n C_n^v s_{i,j-n} \end{cases} \quad (4)$$

Where  $C_n^v$  is the coefficient,  $n \geq 3$  is an integer constant and  $\Gamma$  is the gamma function,

$$C_n^v = \frac{\Gamma(v+1)}{\Gamma(n+1)\Gamma(v-n+1)} \quad (5)$$

Algorithm 2 [21] implements Tiansi fractional differential gradient mask 5x5. The mask coefficients of the fractional differential operator are shown in equation 6:

$$C_{s_n} = (-1)^n \frac{\Gamma(v+1)}{n! \Gamma(v-n+1)} \quad (6)$$

Algorithm 3 is an improved covering template of the fractional differential on x or y coordinates by using the G-L definition of fractional calculus, a generalized fractional-order filter, and modified the coefficient of  $-v$  to be  $1/5$  and that of  $v^2-v$  to be  $1/6$ , was presented in [22].

Algorithm 4 [23] proposed the combination of fractional-order edge detection (FOED) and a chaos synchronization classifier for fingerprint identification. It is based on the G-L definition. In order to overcome the limitations of the integral-order method, FOED has been improved fingerprint images with the clarity of the ridge and valley structures.

Then, in the other four algorithms (from algorithm 5 to 8) use the genetic algorithm with the fractional edge detection to enhance the selection of the threshold levels of the edge detection techniques for each image automatically.

First, read the input image, then apply a fully automatic way to cluster an image using K-means principle has been applied to different fractional edge detection algorithms to extract required number of thresholds. After that, use the different fractional masks algorithms that used previously with different fractional orders to get the output image. And finally, use the genetic algorithm by initializing the population randomly, evaluating the fitness and then applying the selection, crossover, and mutation. Fig.2 shows a flowchart for the genetic algorithm using an automatic number of thresholds with different fractional masks. Table 4 shows that genetic algorithm gets the optimal thresholds for each image with different types.

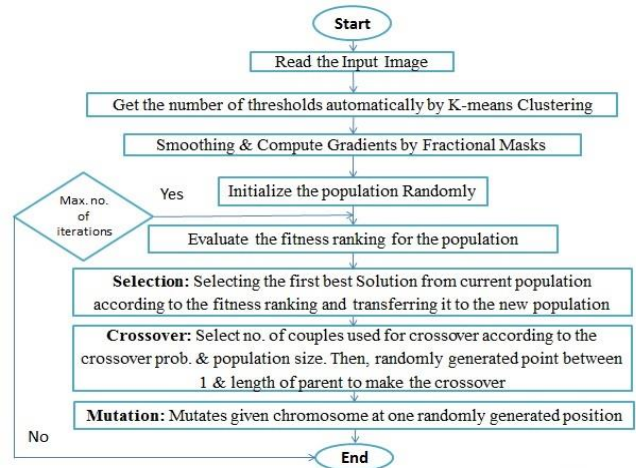


Figure 2: Genetic Algorithm

### 3 Results and Discussion

The algorithms have been implemented in MATLAB using different types of Images. To ensure that the algorithms can work on different types of images, these algorithms are applied on medical images like MRI (Magnetic Resonance Image) and x-ray images. Table 5 shows the images that used the different fractional masks without optimization of thresholds. And table 6 shows the images use the genetic algorithm with different fractional edge detection to enhance the selection of the optimal thresholds.

The performance comparison is done by measuring the MSE & PSNR, bit error ratio and the execution time. The PSNR is measured between the noise free image and the noisy image (salt & pepper noise) with noise density=0.02.

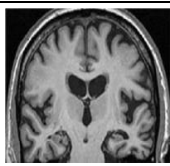
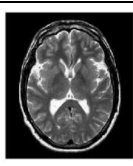




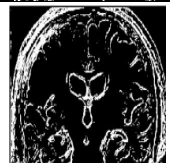


Table 3: Fractional masks and equations used in the different fractional edge detection algorithms

Algorithm	Equations	Parameters	Masks																																																			
			X-direction	Y-direction																																																		
1	$O_x^v = \frac{1}{2} \left( \frac{\partial^v s(x+1, y-1)}{\partial x^v} + 2 \frac{\partial^v s(x+1, y)}{\partial x^v} + \frac{\partial^v s(x+1, y+1)}{\partial x^v} \right)$ $O_y^v = \frac{1}{2} \left( \frac{\partial^v s(x-1, y+1)}{\partial y^v} + 2 \frac{\partial^v s(x, y+1)}{\partial y^v} + \frac{\partial^v s(x+1, y+1)}{\partial y^v} \right)$	V: fractional order	<table border="1"> <tr> <td><math>(-1)^n C_n^v / 2</math></td> <td><math>(-1)^n C_n^v</math></td> <td><math>(-1)^n C_n^v / 2</math></td> </tr> <tr> <td>.</td> <td>.</td> <td>.</td> </tr> <tr> <td>..</td> <td>.</td> <td>.</td> </tr> <tr> <td><math>(v^2-v)/4</math></td> <td><math>(v^2-v)/2</math></td> <td><math>(v^2-v)/4</math></td> </tr> <tr> <td><math>-v/2</math></td> <td><math>-v</math></td> <td><math>-v/2</math></td> </tr> <tr> <td><math>1/2</math></td> <td><math>1</math></td> <td><math>1/2</math></td> </tr> </table>	$(-1)^n C_n^v / 2$	$(-1)^n C_n^v$	$(-1)^n C_n^v / 2$	.	.	.	..	.	.	$(v^2-v)/4$	$(v^2-v)/2$	$(v^2-v)/4$	$-v/2$	$-v$	$-v/2$	$1/2$	$1$	$1/2$	<table border="1"> <tr> <td><math>(-1)^n C_n^v / 2</math></td> <td>...</td> <td><math>(v^2-v)/4</math></td> <td><math>-v/2</math></td> <td><math>1/2</math></td> </tr> <tr> <td><math>(-1)^n C_n^v</math></td> <td>...</td> <td><math>(v^2-v)/2</math></td> <td><math>-v</math></td> <td><math>1</math></td> </tr> <tr> <td><math>(-1)^n C_n^v / 2</math></td> <td>...</td> <td><math>(v^2-v)/4</math></td> <td><math>-v/2</math></td> <td><math>1/2</math></td> </tr> </table>	$(-1)^n C_n^v / 2$	...	$(v^2-v)/4$	$-v/2$	$1/2$	$(-1)^n C_n^v$	...	$(v^2-v)/2$	$-v$	$1$	$(-1)^n C_n^v / 2$	...	$(v^2-v)/4$	$-v/2$	$1/2$																	
$(-1)^n C_n^v / 2$	$(-1)^n C_n^v$	$(-1)^n C_n^v / 2$																																																				
.	.	.																																																				
..	.	.																																																				
$(v^2-v)/4$	$(v^2-v)/2$	$(v^2-v)/4$																																																				
$-v/2$	$-v$	$-v/2$																																																				
$1/2$	$1$	$1/2$																																																				
$(-1)^n C_n^v / 2$	...	$(v^2-v)/4$	$-v/2$	$1/2$																																																		
$(-1)^n C_n^v$	...	$(v^2-v)/2$	$-v$	$1$																																																		
$(-1)^n C_n^v / 2$	...	$(v^2-v)/4$	$-v/2$	$1/2$																																																		
2	$\frac{d^v s(x, y)}{dx^v} \approx s(x, y) + (-v)s(x-1, y) + \frac{v(v-1)}{2} s(x-2, y) + \dots$ $+ (-1)^n \frac{[(v+1)]}{n! [(v-n+1)]} s(x-n, y)$ $\frac{d^v s(x, y)}{dy^v} \approx s(x, y) + (-v)s(x, y-1) + \frac{v(v-1)}{2} s(x, y-2) + \dots$ $+ (-1)^n \frac{[(v+1)]}{n! [(v-n+1)]} s(x, y-n)$		<table border="1"> <tr> <td><math>(v^2-v)/2</math></td> <td><math>-v</math></td> <td><math>0</math></td> <td><math>v</math></td> <td><math>(v^2-v)/2</math></td> </tr> <tr> <td><math>(v^2-v)</math></td> <td><math>-2v</math></td> <td><math>0</math></td> <td><math>2v</math></td> <td><math>(v^2-v)</math></td> </tr> <tr> <td><math>3(v^2-v)/2</math></td> <td><math>-3v</math></td> <td><math>0</math></td> <td><math>3v</math></td> <td><math>3(v^2-v)/2</math></td> </tr> <tr> <td><math>(v^2-v)</math></td> <td><math>-2v</math></td> <td><math>0</math></td> <td><math>2v</math></td> <td><math>(v^2-v)</math></td> </tr> <tr> <td><math>(v^2-v)/2</math></td> <td><math>-v</math></td> <td><math>0</math></td> <td><math>v</math></td> <td><math>(v^2-v)/2</math></td> </tr> </table>	$(v^2-v)/2$	$-v$	$0$	$v$	$(v^2-v)/2$	$(v^2-v)$	$-2v$	$0$	$2v$	$(v^2-v)$	$3(v^2-v)/2$	$-3v$	$0$	$3v$	$3(v^2-v)/2$	$(v^2-v)$	$-2v$	$0$	$2v$	$(v^2-v)$	$(v^2-v)/2$	$-v$	$0$	$v$	$(v^2-v)/2$	<table border="1"> <tr> <td><math>(v^2-v)/2</math></td> <td><math>(v^2-v)</math></td> <td><math>3(v^2-v)/2</math></td> <td><math>(v^2-v)</math></td> <td><math>(v^2-v)/2</math></td> </tr> <tr> <td><math>-v</math></td> <td><math>-2v</math></td> <td><math>-3v</math></td> <td><math>-2v</math></td> <td><math>-v</math></td> </tr> <tr> <td><math>0</math></td> <td><math>0</math></td> <td><math>0</math></td> <td><math>0</math></td> <td><math>0</math></td> </tr> <tr> <td><math>v</math></td> <td><math>2v</math></td> <td><math>3v</math></td> <td><math>2v</math></td> <td><math>v</math></td> </tr> <tr> <td><math>(v^2-v)/2</math></td> <td><math>(v^2-v)</math></td> <td><math>3(v^2-v)/2</math></td> <td><math>(v^2-v)</math></td> <td><math>(v^2-v)/2</math></td> </tr> </table>	$(v^2-v)/2$	$(v^2-v)$	$3(v^2-v)/2$	$(v^2-v)$	$(v^2-v)/2$	$-v$	$-2v$	$-3v$	$-2v$	$-v$	$0$	$0$	$0$	$0$	$0$	$v$	$2v$	$3v$	$2v$	$v$	$(v^2-v)/2$	$(v^2-v)$	$3(v^2-v)/2$	$(v^2-v)$	$(v^2-v)/2$
$(v^2-v)/2$	$-v$	$0$	$v$	$(v^2-v)/2$																																																		
$(v^2-v)$	$-2v$	$0$	$2v$	$(v^2-v)$																																																		
$3(v^2-v)/2$	$-3v$	$0$	$3v$	$3(v^2-v)/2$																																																		
$(v^2-v)$	$-2v$	$0$	$2v$	$(v^2-v)$																																																		
$(v^2-v)/2$	$-v$	$0$	$v$	$(v^2-v)/2$																																																		
$(v^2-v)/2$	$(v^2-v)$	$3(v^2-v)/2$	$(v^2-v)$	$(v^2-v)/2$																																																		
$-v$	$-2v$	$-3v$	$-2v$	$-v$																																																		
$0$	$0$	$0$	$0$	$0$																																																		
$v$	$2v$	$3v$	$2v$	$v$																																																		
$(v^2-v)/2$	$(v^2-v)$	$3(v^2-v)/2$	$(v^2-v)$	$(v^2-v)/2$																																																		
3	$\frac{d^v s(x, y)}{dx^v} \approx \sum_{z=-a}^a \sum_{k=-2b}^0 W_A(z, k) s(x+z, y+k)$ $\frac{d^v s(x, y)}{dy^v} \approx \sum_{z=-2a}^0 \sum_{k=-b}^b W_B(z, k) s(x+z, y+k)$	<p><math>W_A</math> and <math>W_B</math>: covering templates on x and y coordinates respectively. <math>m=2a+1</math> <math>n=2b+1</math> <math>a=(m-1)/2</math> <math>b=(n-1)/2</math></p>	<table border="1"> <tr> <td><math>(v^2-v)/6</math></td> <td><math>(v^2-v)/6</math></td> <td><math>(v^2-v)/6</math></td> </tr> <tr> <td><math>-v/5</math></td> <td><math>-v/5</math></td> <td><math>-v/5</math></td> </tr> <tr> <td><math>-v/5</math></td> <td><math>1</math></td> <td><math>-v/5</math></td> </tr> </table>	$(v^2-v)/6$	$(v^2-v)/6$	$(v^2-v)/6$	$-v/5$	$-v/5$	$-v/5$	$-v/5$	$1$	$-v/5$	<table border="1"> <tr> <td><math>(v^2-v)/6</math></td> <td><math>-v/5</math></td> <td><math>-v/5</math></td> </tr> <tr> <td><math>(v^2-v)/6</math></td> <td><math>-v/5</math></td> <td><math>1</math></td> </tr> <tr> <td><math>(v^2-v)/6</math></td> <td><math>-v/5</math></td> <td><math>-v/5</math></td> </tr> </table>	$(v^2-v)/6$	$-v/5$	$-v/5$	$(v^2-v)/6$	$-v/5$	$1$	$(v^2-v)/6$	$-v/5$	$-v/5$																																
$(v^2-v)/6$	$(v^2-v)/6$	$(v^2-v)/6$																																																				
$-v/5$	$-v/5$	$-v/5$																																																				
$-v/5$	$1$	$-v/5$																																																				
$(v^2-v)/6$	$-v/5$	$-v/5$																																																				
$(v^2-v)/6$	$-v/5$	$1$																																																				
$(v^2-v)/6$	$-v/5$	$-v/5$																																																				
4	$\frac{d^v s(x, y)}{dx^v} \approx s(x, y) + (-v)s(x-1, y)$ $+ \frac{(-v)(-v+1)}{2} s(x-2, y) + \dots$ $+ \frac{[(-v+1)]}{n! [(-v+n+1)]} s(x-n, y)$ $\frac{d^v s(x, y)}{dy^v} \approx s(x, y) + (-v)s(x, y-1)$ $+ \frac{(-v)(-v+1)}{2} s(x, y-2) + \dots$ $+ \frac{[(-v+1)]}{n! [(-v+n+1)]} s(x, y-n)$	V: fractional order	<table border="1"> <tr> <td><math>0</math></td> <td><math>(v^2-v)/2</math></td> <td><math>0</math></td> </tr> <tr> <td><math>0</math></td> <td><math>-v</math></td> <td><math>0</math></td> </tr> <tr> <td><math>0</math></td> <td><math>1</math></td> <td><math>0</math></td> </tr> </table>	$0$	$(v^2-v)/2$	$0$	$0$	$-v$	$0$	$0$	$1$	$0$	<table border="1"> <tr> <td><math>0</math></td> <td><math>0</math></td> <td><math>0</math></td> </tr> <tr> <td><math>(v^2-v)/2</math></td> <td><math>-v</math></td> <td><math>1</math></td> </tr> <tr> <td><math>0</math></td> <td><math>0</math></td> <td><math>0</math></td> </tr> </table>	$0$	$0$	$0$	$(v^2-v)/2$	$-v$	$1$	$0$	$0$	$0$																																
$0$	$(v^2-v)/2$	$0$																																																				
$0$	$-v$	$0$																																																				
$0$	$1$	$0$																																																				
$0$	$0$	$0$																																																				
$(v^2-v)/2$	$-v$	$1$																																																				
$0$	$0$	$0$																																																				

Table 4: Optimal thresholds

Algorithm	Alzheimer Brain (383x270)	MRI Brain (314x348)	Xray Hand (645x1024)
GA Sobel ED	3, 16, 19	47, 128, 188, 254	6, 9, 11, 12
Algo. 5 (v=0.2)	22, 56, 95	28, 63, 109, 140	74, 100, 130, 161
Algo. 5 (v=0.8)	23, 77, 135	50, 100, 160, 204	104, 142, 180, 234
Algo. 6 (v=0.2)	39, 67, 95	68, 129, 191, 254	89, 127, 168, 207
Algo. 6 (v=0.8)	102, 176, 255	20, 36, 44, 65	96, 141, 195, 249
Algo. 7 (v=0.2)	32, 80, 140	48, 97, 166, 216	112, 160, 208, 239
Algo. 7 (v=0.8)	7, 17, 115	11, 58, 88, 118	18, 28, 37, 164
Algo. 8 (v=0.2)	27, 80, 140	51, 108, 159, 206	107, 149, 191, 231
Algo. 8 (v=0.8)	8, 15, 162	13, 63, 102, 135	3, 8, 159, 185

Table 5: Different types of Images using different Fractional algorithms

Algorithm	Alzheimer Brain (383x270)	MRI Brain (314x348)	XrayHand (645x1024)
Original Image			
Edge Detection			
Algo.1 (v=0.2)			

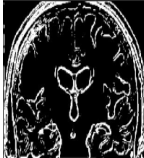
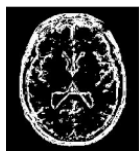

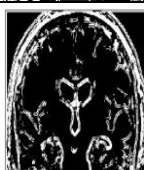


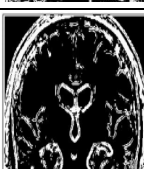


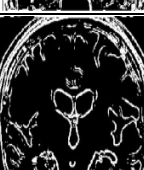


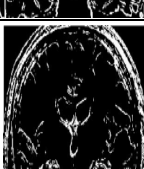
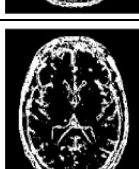

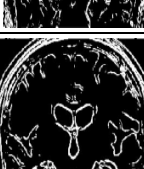
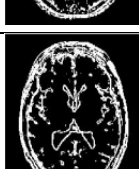

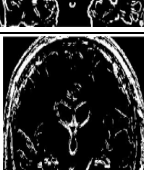
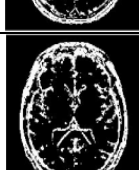


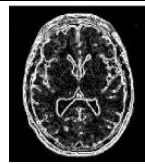


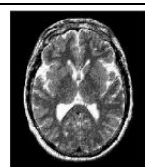

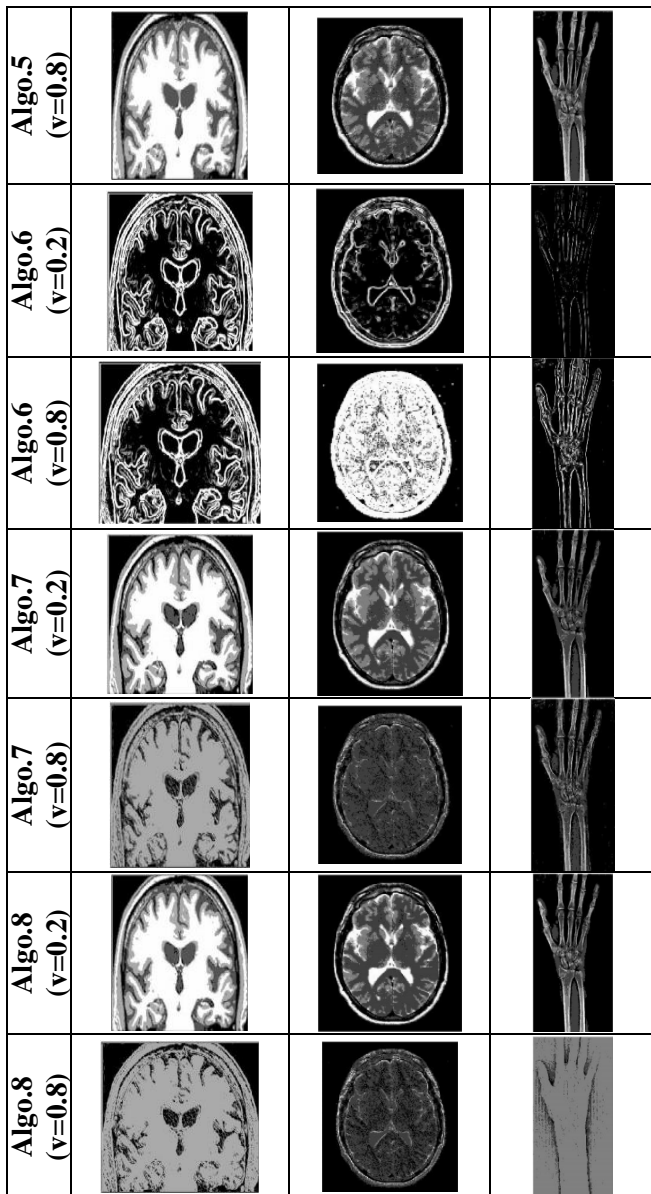
Algo.1 (v=0.8)			
Algo.2 (v=0.2)			
Algo.2 (v=0.8)			
Algo.3 (v=0.2)			
Algo.3 (v=0.8)			
Algo.4 (v=0.2)			
Algo.4 (v=0.8)			

Table 6: Different types of Images using Genetic Algorithm with integer and Fractional edge detection

Algorithm	Alzheimer Brain (383x270)	MRI Brain (314x348)	Xray Hand (645x1024)
GA Sobel ED			
Algo.5 (v=0.2)			





The execution time for images depends first on the dimension of each image, then the algorithm used. Fig. 3 shows the diagram for the execution time. We noticed that algorithms 7 & 8 take greater time but it gives the best results.

According to the MSE in Table 7, the highest result in Alzheimer Brain is algorithm 8 when  $v=0.2$ , MRI Brain, and XrayHand is in algorithm 7 when  $v=0.2$ . The PSNR results in Table 7, the highest result in Alzheimer Brain is algorithm 8 when  $v=0.2$ , MRI Brain, and XrayHand is in algorithm 7 when  $v=0.2$ . And the bit error rate in Table 7, the best algorithm is algorithm 7 when  $v=0.2$  for MRI Brain, and XrayHand images. Alzheimer Brain image has best results in algorithm algorithm 8 when  $v=0.2$ .

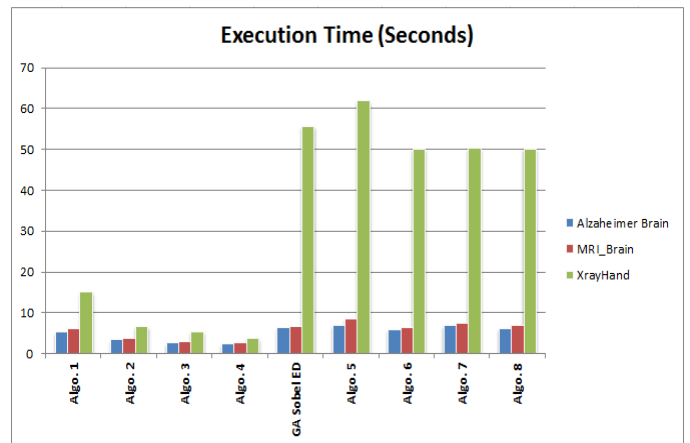


Figure 3: Execution Time in seconds for Different Images

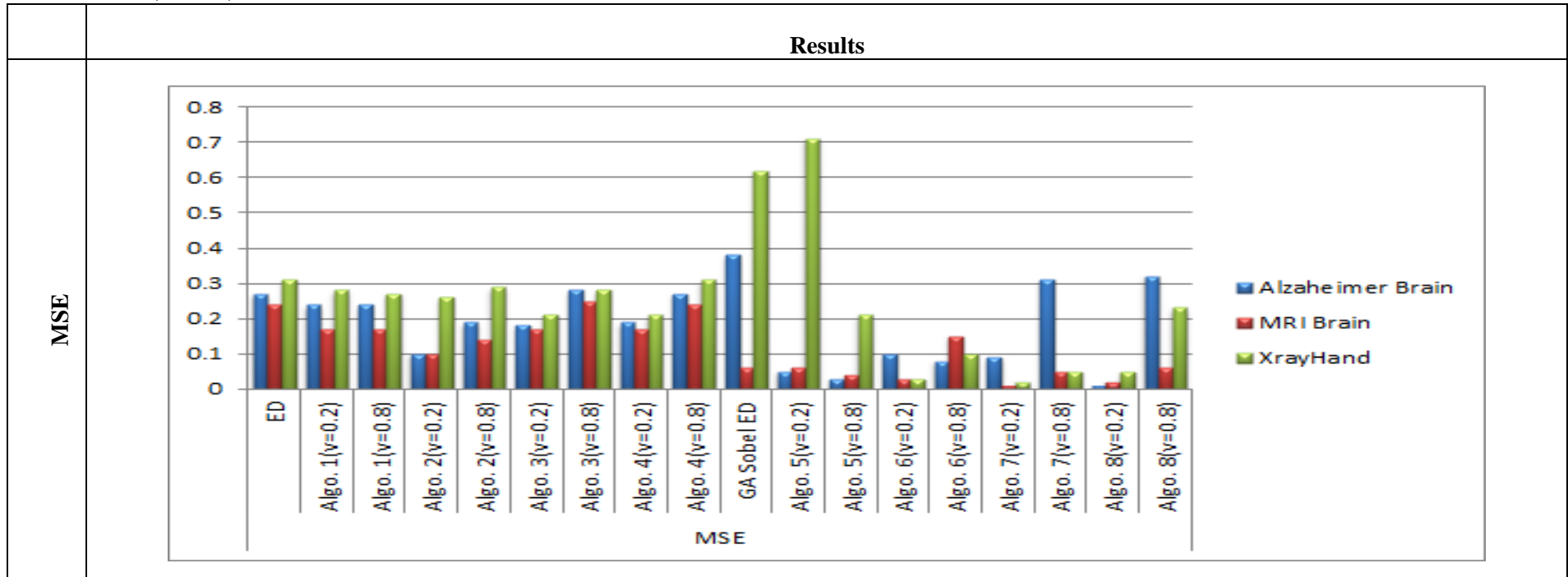
## 4 Conclusion

Nowadays, to enhance the edge information effectively, many research papers use of fractional-order differential methods with edge detection operators, but without optimization of threshold levels of the algorithm for each image. This paper makes a performance comparison between the fractional edge detection with and without genetic algorithm. From the results, it shows that the fractional-order enhances the performance than the integer-order differentiation and genetic algorithm with fractional edge detection enhances the selection of the threshold levels of the edge detection techniques for each image automatically. From the PSNR and bit error rate results, it can be concluded that the best fractional algorithms based on the genetic algorithm can be applied to different types of images are GA Fractional\_Mask2 and GA Fractional\_Mask3 when  $v=0.2$ .

## 5 Acknowledgments

The authors would like to express their greeting for Nile University, Dr. Lobna Said for her support during experimental work.

Table 7: MSE, PSNR, BER Results







*References:*

- [1] S.M. Ismail, A.G. Radwan, A.H. Madian, M.F. Abu-ElYazeed, Comparative Study of Fractional Filters for Alzheimer Disease Detection on MRI images, *39<sup>th</sup> International Conference on Telecommunications and Signal Processing (TSP), IEEE*, 2016, pp.720-723.
- [2] L.K. Lee, S.C. Liew, A Survey of Medical Image Processing Tools, *4<sup>th</sup> International Conference on Software Engineering and Computer Systems (ICSECS), IEEE, Malaysia*, 2015, pp. 171-176.
- [3] R. Yogamangalam, B. Karthikeyan, Segmentation Techniques Comparison in Image Processing, *International Journal of Engineering and Technology (IJET)*, Vol.5, No.1, 2013, pp. 307-313.
- [4] H.P. Narkhede, Review of Image Segmentation Techniques, *International Journal of Science and Modern Engineering (IJISME)*, Vol.1, No.8, 2013, pp. 54-61.
- [5] S. Patra, R. Gautam, A. Singla, A novel context sensitive multilevel thresholding for image segmentation, *Applied Soft Computing* 23, 2014, pp. 122–127.
- [6] S. Nilima, P. Dhanesh, J. Anjali, Review on Image Segmentation, Clustering and Boundary Encoding, *International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)*, Vol.2, No.11, 2013, pp. 6309-6314.
- [7] K.K. Rahini, S.S. Sudha, Review of Image Segmentation Techniques: A Survey, *International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE)*, Vol.4, No.7, 2014, pp. 842-845.
- [8] B.L. Srinivas, Hemalatha, K.A. Jeevan, Edge Detection Techniques for Image Segmentation, *International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)*, Vol.3, 2015, pp. 288-292.
- [9] D. Tian, J. Wu, Y. Yang, A Fractional-order Edge Detection Operator for Medical Image Structure Feature Extraction, *26<sup>th</sup> Chinese Control and Decision Conference (CCDC), IEEE*, 2014, pp. 5173-5176.
- [10] J. Mehena, M.C. Adhikary, Medical Image Edge Detection Based on Soft Computing Approach, *International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)*, Vol.3, No.7, 2015, pp. 6801-6807.
- [11] N.S. Joshi, N.S. Choubey, Application of Soft Computing Approach for Edge Detection, *International Journal of Application or Innovation in Engineering & Management (IJAEM)*, Vol.3, No.4, 2014, pp. 116-122.
- [12] S. Das, *Functional Fractional Calculus*, Springer-Verlag, ch.1, 2011.
- [13] E.A. Gonzalez, I. Petráš, Advances in Fractional Calculus: Control and Signal Processing Applications, *16<sup>th</sup> International Carpathian Control Conference (ICCC), IEEE, Szilvasvarad, Hungary*, 2015, pp. 147 -152.
- [14] C. Gao, J. Zhou, W. Zhang, Edge Detection Based on the Newton Interpolation's Fractional Differentiation, *The International Arab Journal of Information Technology*, Vol.11, No.3, 2014, pp. 223-228.
- [15] 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, 2010, pp. 491-511.
- [16] P. Ghosh, Medical Image Segmentation Using a Genetic Algorithm, *Ph.D. dissertation, Department Electrical and Computer Engineering, Portland State University*, 2010.
- [17] A.P. Engelbrecht, *Computational Intelligence: an Introduction*, 2<sup>nd</sup> Edition. Hoboken, NJ: Wiley Publishing, 2007.
- [18] A.A. Funmilola, O.A. Oke, T.O. Adedeji, O.M. Alade, E.A. Adewusi, Fuzzy k-c-means clustering algorithm for medical image segmentation, *Journal of Information Engineering and Applications*, Vol.2, No.6, 2012, pp. 21–32.
- [19] W.S. ElAraby, A.H. Median, M.A. Ashour, I. Farag, M. Nassef, Fractional canny edge detection for biomedical applications, *28<sup>th</sup> International Conference on Microelectronics (ICM), IEEE*, 2016, pp. 265-268.
- [20] D. Tian, J. Wu, Y. Yang, A Fractional-order Edge Detection Operator for Medical Image Structure Feature Extraction, *26<sup>th</sup> Chinese Control and Decision Conference (CCDC)*, 2014, pp. 5173-5176.
- [21] Z. Yang, F. Lang, X. Yu, Y. Zhang, The Construction of Fractional Differential Gradient Operator, *Journal of Computational Information Systems*, 2011, pp. 4328-4342.
- [22] Y. Pu, Fractional calculus approach to texture of digital image, *IEEE International Conference on Signal Processing*, 2006, pp. 1002–1006.
- [23] J. Chen, C. Huang, Y. Du, C. Lin, Combining fractional-order edge detection and chaos synchronisation classifier for fingerprint identification, *IET Image Processing*, Vol.8, No.6, 2014, pp. 354–362.