A New hybrid Method for Noise Robust Estimation of Image Fractal Dimension

Saviz Ebrahimi[1], Farbod Setoudeh[2*], Mohammad Bagher Tavakoli[1]
1- Department of Electrical Engineering, Arak Branch, Islamic Azad University, Arak, Iran.
Email: s-ebrahimi92@iau-arak.ac.ir
Email: m-tavakoli@iau-arak.ac.ir
2- Department of Electrical Engineering, Arak University of Technology, Arak, Iran.
Email: F.setoudeh@arakut.ac.ir (Corresponding author)

Received: September 2019 Revised: November 2019 Accepted: January 2020

ABSTRACT:
This paper presents a modified model to calculate the fractal dimension of digital images. The estimation of fractal dimensions is crucial to fractal analysis and is popularly carried out through methods based on box counting. The problem with these approaches is that, most of them do not remove the potential effects of noise on fractal dimensions properly. Accordingly, this study examines the effects of three different type of noises on fractal dimensions using different images taken from Background image database. The examination shows that the fractal dimensions change significantly, after noise adding, so we put forward a noise-robust and efficient fractal dimension calculation method which is a combination of two methods, the gray-level co-matrix algorithm and improved box counting method. The results of experiments on the Background image dataset confirm the robustness and efficiency of the proposed method.

KEYWORDS: Box-Counting Algorithm, Fractal Dimension, Gray-Level Co-Matrix, Image, Noise.

1. INTRODUCTION

A fractal is a mathematical set and an irregular geometric object that models a repeating pattern displayed at every scale. Mandelbrot was the first to introduce fractals in the estimation of surface or texture roughness [1]. Fractal geometry provides an appropriate mathematical method of studying the irregular and complex forms found in nature. It is used to analyze the physical structural irregularities that may not be, in general, represented by Euclidean geometry [2]. Most gray-level images of nature have fractal characteristics [3], and the fractal properties of real-world objects are commonly examined in digital images. The most important parameter of a fractal is its fractal dimension, which plays a critical role in texture analysis, image segmentation, shape classification, feature extraction, computer vision, and medical image analysis. Thus, the accurate calculation of fractal dimensions is crucial to these applications. By using texture analysis, the similarities in the estimation of textures and fractal features are distinguished and popularized [4].

Many techniques for fractal dimension estimation have been proposed. For example, [5] proposes four methods to estimate fractal dimensions from surface descriptions and the applicability of fractals to measure surfaces. The researchers first used generated surfaces to analyze the methods and then later employed the approaches to measured surfaces. The fractal dimension values of all generated descriptions were computed by box-counting, power spectral density and roughness-length. The fractal dimension estimation method proposed in [6] for RGB color images is an expansion of the DBC algorithm, in which a counting approach that is feasible for RGB color images was incorporated. This method presents a hyper-surface partition strategy, which regards a hyper-surface as a continuous element and divides an image into non-overlapping blocks. In [7], the researchers put forward a new conceptual fractal dimension calculation method that is especially suitable for curves. The approach is based on the novel concept of induced fractal structure on an image set of any curve. Some theoretical properties of this new denotation of fractal dimensions and a result that allows the construction of space-filling curves were discussed. A method of estimating the fractal dimensions and fractal curvatures of binary digital images was proposed in [8]. The method includes an analysis of several geometric characteristics, such as the intrinsic volumes of the parallel sets of a fractal.

The DBC method is unsuitable for low-resolution images because the existence of empty boxes influences the accuracy with which fractal dimensions are estimated. Correspondingly, the study in [9] presented a new algorithm called actual DBC, which classifies empty boxes into real empty boxes and potential empty boxes. Associating the spatial domain relationships between a fractional Brownian surface model and a pixel gray level enables the calculation of the probability that empty boxes will become potential ones under high resolutions. Even under an insufficiently high image resolution, the method still enables an accurate estimation of fractal dimensions.

Among the different methods presented in the literature, a commonly used approach is the grid dimension method, which is popularly known as the box-counting method of estimating fractal dimensions. Among different box-counting algorithms, DBC covers a wide dynamic range and efficiently computes fractal dimensions; it is commonly used to calculate the fractal dimensions of 2D gray-level images. The popularity of DBC stems from its simplicity and automatic computability [10]. The DBC method proposed by Sarkar and Chaudhuri [11–13] has been applied in many studies that focus on gray-level images.

Despite the advantages of DBC, however, it is easily affected by noise. A few studies have analyzed the effects of noise and developed methods of eliminating this problem. For instance, an anti-noise method based on the DBC algorithm was proposed in [14]. The method takes full advantage of each pixel in a box and replaces the extremum deviation with standard deviation. A comparison of the anti-noise DBC algorithm and the conventional DBC algorithm shows that the former generates a better fractal dimension value in many cases.

The above-mentioned methods are useful approaches, but they are not robust and stable under increasing noise variance or density. In general, images are inevitably affected by noises during collection and communication, thereby also affecting the quality the fractal dimensions of images are changed. One approach to ensure the robustness and stability is to first remove noise and then conduct image processing. However, this strategy is not effective enough and also preprocessing is not possible every time. A good strategy would be to apply directly to noisy digital images. Implementing this strategy strongly requires identifying the effects of noise on digital images while estimating their fractal dimensions and designing a method that is robust against noise.

In the current work, a robust and efficient method for calculating fractal dimensions is proposed. This method is a combination of two methods, the Gray-Level Co-Matrix (GLCM) algorithm and improved box counting method. Natural texture images obtained from the Background image dataset is used to validate the performance of the method proposed in the present research.

2. GRAY-LEVEL CO-OCCURRENCE MATRIX

A GLCM is a matrix that is defined over an image to be the distribution of co-occurring pixel values. A statistical technique, such as the co-occurrence matrix, facilitates the collection of useful information about the positions of neighboring pixels in a texture image. To generate a GLCM, a set of offsets sweeping through 180 degrees at the same distance parameter D are used to achieve a degree of rotational invariance (*[0 D]* for *0°*: P horizontal, *[-D D]* for *45°*: P right diagonal, *[-D 0]* for *90°*: P vertical, and *[-D -D]* for *135°*: P left diagonal). Fig. 1 illustrates GLCM generation in four directions [24]. The number of rows and columns in a GLCM is equal to the number of gray levels in an image. The use of numerous intensity levels implies the storage of considerable temporary data for each combination, thus, the number of gray levels is often reduced [17].

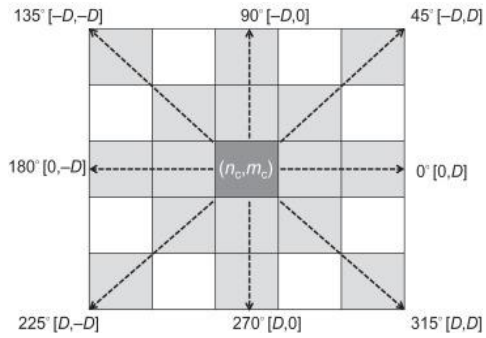

Fig. 1. GLCM generation.

Haralick [18] introduced 14 statistical features that are generated by calculating the features for each co-occurrence matrix obtained. The features are calculated using the directions *0°*, *45°*, *90°*, and *135°*, after which these four values are averaged. Contrast, homogeneity, angular second moment (energy), and correlations are the features that these measurements define in Equations (1) - (4), respectively, μ_i and δ_i are defined by (5) and (6). Contrast quantifies variations in image intensity, providing a measure of gray-level contrast between neighboring pixels over an entire image. Angular Second Moment (ASM), which is also known as uniformity or energy, is a measure of texture uniformity in gray-level spatial distribution. Homogeneity reflects the heterogeneity of a texture pattern and decreases with contrast [19]. Correlation is a measure of gray-level linear dependence between pixels at specified positions relative to each other. In Equations (15-20), N_g is the total number of gray levels and *P(i, j)* is the GLCM of an image.

$$Contrast = \sum_{i=0}^{N_{g-1}} \sum_{j=0}^{N_{g-1}} (i-j)^2 \times p(i,j) \quad (1)$$

$$Homogeneity = \sum_{i=0}^{N_{g-1}} \sum_{j=0}^{N_{g-1}} \frac{p(i,j)}{1+(i-j)^2} \quad (2)$$

$$ASM = \sum_{i=0}^{N_{g-1}} \sum_{j=0}^{N_{g-1}} p(i,j)^2 \quad (3)$$

$$Correlation = \sum_{i=0}^{N_{g-1}} \sum_{j=0}^{N_{g-1}} \frac{(i-\mu_i)(j-\mu_j)p(i,j)}{\delta_i \delta_j} \quad (4)$$

$$\mu_i = \sum_{i=0}^{N_{g-1}} \sum_{j=0}^{N_{g-1}} i \times p(i,j) \quad (5)$$

$$\delta_i^2 = \sum_{i=0}^{N_{g-1}} \sum_{j=0}^{N_{g-1}} (i-\mu_i)^2 \times p(i,j) \quad (6)$$

3. METHODS OF ESTIMATING FRACTAL DIMENSION

Different approaches can be used to estimate fractal dimensions, and most of these are based on the original DBC. This section reviews some of these methods.

The basic method of estimating fractal dimensions is based on the concept of self-similarity. In order to define the fractal dimension *D* by self-similarity concept, consider a bounded set *A* in Euclidean n-space. The set is said to be self-similar when A is the union of *N*, distinct (no overlapping) copies of itself each of which is similar to *A* scaled down by a ratio ***r.*** Fractal dimension *D* of A can be defined as (7).

$$\mathrm{D} = \frac{\log(\mathrm{N_r})}{\log(1/\mathrm{r})} \quad (7)$$

The fractal dimension can be calculated only for deterministic fractals, and the fractal dimension of an object with deterministic self-similarity is equal to its box-counting dimension. This method finds limited application because natural scenes or natural fractals are of a non-ideal and non-deterministic character [12]. The original DBC is described as follows [15].

Consider an image of size $M\times M$ as a three-dimensional image having a surface with *(x, y)* as positions; the third coordinate, *z* indicates a pixel gray level. In the DBC method, the image surface is partitioned into non-overlapping blocks of size $s\times s$. The scale of each block is $r = s/M$, where $2\leq s\leq M/2$, and s is an integer. N_r denotes the total number of boxes that cover the image in all scales and is calculated thus: let us suppose that a column of boxes of size $s\times s\times s'$ exists on each block, where s' is the height of each box and is defined as $G/s' = M/s$. *G* represents the total number of gray levels. If the minimum and maximum gray-level in the (i, j)th block fall inbox number k and l, respectively, the number of boxes covering this block is calculated using (8).

$$n_r(i,j) = L - K + 1 \quad (8)$$

Where *r* denotes the scale and N_r is counted by (9).

$$N_r = \sum n_r(i,j) \quad (9)$$

Then, the fractal dimension can be estimated from the least squares linear fit of *log* (N_r) versus *log (1/r)* [15]. Fig. 2 depicts the determination of the number of boxes by the DBC method.

Jin et al. [16] proposed a relative DBC (RDBC) method, in which N_r is computed by (10).

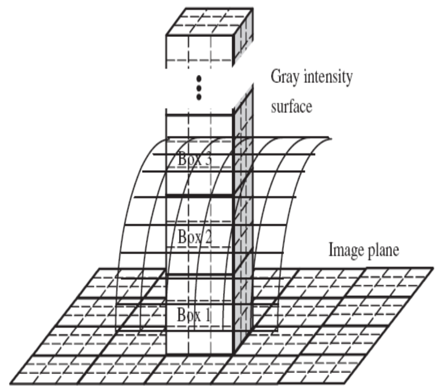

Fig. 2. Determination of the number of boxes by the DBC method [15].

$$N_r = \sum ceil[d_r(i\ ,j)]/s' \quad (10)$$

Where, $d_r(i, j)$ is defined as Equation (11). In (11), I_{max} and I_{min} are the maximum and minimum values of intensity in the *(i,j)*th block, respectively; *ceil [ ]* denotes the ceiling function; and *s'* represents the height of each box. The fractal dimension can be estimated from the least squares linear fit of *log* (N_r) versus *log (1/r).*

$$d_r(i,j) = I_{max} - I_{min} \quad (11)$$

Li et al. [15] developed an algorithm on the basis of the DBC method to calculate the fractal dimensions of gray-level images. If the maximum and minimum gray levels of the *(i, j)*th block are *l* and *k*, respectively, the number of boxes that cover each block, n_r is calculated by (12).

$$n_r(i,j) = \begin{cases} Ceil\left(\frac{l-k}{r'}\right) & l \neq k \\ 1 & l = k \end{cases} \quad (12)$$

Where, *r'* is the box height and is calculated using Equation (13), which *a* is a positive integer and σ represents the mean and standard deviation of a digital image. While r is defined as r=s/M, s and M are the size of each block and the image respectively. The total number of boxes that cover an entire image is computed using Equation (9). As with the process in the above-mentioned approaches, the fractal dimension can be estimated from the least squares linear fit of *log* (N_r) versus *log (1/r).*

$$r' = \frac{r}{1+2a\delta} \tag{13}$$

Juan et al. [14] used the average gray level within the *(i, j)*th block to compute fractal dimensions. Here, $n_r(i, j)$ is defined by (14).

$$n_r(i,j) = \begin{cases} 1 + w.(\overline{g} - g_{min}) & (\overline{g} - g_{min}) > (g_{max} - \overline{g}) \\ 1 + w.(g_{max} - \overline{g}) & else \end{cases} \tag{14}$$

Where, w is the weight, presented in Equation (15); g_{min}, g_{max} and $\bar{g}$ are the minimum, maximum and average gray-level respectively in the *(i, j)*th block.

$$w = \begin{cases} N/_{N_{min}} & (\overline{g} - g_{min}) > (g_{max} - \overline{g}) \\ N/_{N_{max}} & else \end{cases} \tag{15}$$

N is the total number of pixels within a block; N_{min} is the number of pixels with a value falling between the minimum and mean values of the gray level; and N_{max} represents the number of pixels with a value that falls between the mean value and maximum values of the gray level. The total number of boxes covering an image plane is computed using Equation (9), and the fractal dimension can be estimated from the least squares linear fit of $log\ (N_r)$ versus $log\ (1/r)$.

4. PROPOSED METHOD FOR FRACTAL DIMENSION CALCULATION

Given that the original DBC algorithm calculates the difference between the maximum and minimum gray levels to determine the number of boxes in each block, under negligible noise, such number clearly changes.

As previously stated, the critical parameter in fractal geometry is fractal dimension, whose accurate estimation is very important in texture analysis. An inevitable challenge encountered in texture analysis and gray level images is the effect of different noises. The effects of noise on the fractal dimensions of images vary, and estimated fractal dimensions' increase because the roughness of an image and the intensity difference between pixels also increase. A fractal dimension is therefore also a parameter that shows an increase in the complexity moment of a fractal object. In other words, the fractal dimension of a noisy image is mostly larger than that of the original image without noise. The original DBC algorithm applies the difference between the maximum and minimum gray levels to calculate the number of boxes in each block. Thus, the intensity of other pixels is minimally affected when the number of boxes in a block is counted. When negligible noise is added to an image, the gray values of some pixels and the maximum and minimum gray levels change. Correspondingly, the number of boxes in each block and the total number of boxes that cover an entire image also change. This is the reason why fractal dimensions vary and deviate from exact values. That is, the DBC algorithm and its modified versions are not immune to noise.

To solve this problem, if the parameters such as mean, variance, or standard deviation are used alternatively, the maximum and minimum gray level difference in the methods based on the DBC is replaced; the noise effect can be eliminated as acceptable. Since in this way, the entire gray levels of the image pixels interact in the counting of the boxes, not just the maximum and minimum values given the definitions and statistical features are derived from the GLCM, this matrix can be useful for providing a noise robust method based on box counting. With regard to the above, we put forward a noise-robust DBC and GLCM-based method for estimating fractal dimensions. The specific improvement presented by the proposed method is point-of-view de-noising with no image preprocessing.

The process of the proposed noise-robust algorithm for fractal dimension calculation is explained as follows.

1) Consider an image of size $M\times M$.
2) Generate the GLCM, with distance equal to 1 ($D=1$) and obtain matrices p_0, $p_{\pi/4}$, $p_{\pi/2}$, and $p_{3\pi/4}$, in four directions 0°, 45°, 90°, and 135° respectively, for the desired image.
3) Calculate the contrast feature of each of the matrices using Equations (16) to (19).

$$C_0 = \sum_{i=0}^{N_{g-1}} \sum_{j=0}^{N_{g-1}} (i-j)^2 \times p_0(i,j) \tag{16}$$

$$C\pi/_4 = \sum_{i=0}^{N_{g-1}} \sum_{j=0}^{N_{g-1}} (i-j)^2 \times p\pi/_4(i,j). \tag{17}$$

$$C\pi/_2 = \sum_{i=0}^{N_{g-1}} \sum_{j=0}^{N_{g-1}} (i-j)^2 \times p\pi/_2(i,j). \tag{18}$$

$$C_{3\pi}/_4 = \sum_{i=0}^{N_{g-1}} \sum_{j=0}^{N_{g-1}} (i-j)^2 \times p_{3\pi}/_4(i,j). \tag{19}$$

4) Generate the C matrix.

$$C = \left[c_0\ c_{\frac{\pi}{4}}\ c_{\frac{\pi}{2}}\ c_{\frac{3\pi}{4}}\right]. \tag{20}$$

5) Calculate the standard deviation of the C matrix (δ_c).

The amount of C in the noisy image is slightly changed compared to the value of C in the original image. Therefore, we can use the standard deviation of the matrix C to count the boxes of each block.

6) Calculate the standard deviation of the desired image (δ_I).
7) Divide the image into non overlapping blocks of size $s\times s$, which is an integer between *2* and *M/2*.
8) Calculate the number of boxes that cover each block, using Equation (21).

$$n_r = 2a \times \left[\frac{\delta_c}{h}\right] + 1. \tag{21}$$

In Equation (21), a is a positive integer whose optimum value is equal to 4. This optimum value is determined by the trial and error and also the results of the examination. Although fractal dimensions for values less than or greater than a = 4 are within acceptable range, but these values, reduce the accuracy and increase the fit error (E), so the optimal value occurs at a = 4. Moreover, in Equation (21), the value of 1 is added to the relationship to prevent zeroing n_r and failing to compute the fractal dimension. Since in the special case if the original image has only one gray level, the contrast of the GLCM is the same in four directions, and therefore δ_c is zero.

In Equation (21), h is the height of each box in order to determine it, according to the contrast values obtained from GLCM, the value of δ_c is small and less than one, so the height of each box should be proportional to Equation (21).

Suppose that most pixels fall into the interval of gray levels within $[\mu - b\sigma_I, \mu + b\sigma_I]$, in which b is a positive integer and μ and σ represent the mean and standard deviation of the image, respectively. Under this condition, a box with a small height is chosen for an image surface with high intensity variations. The height of boxes in this method is much smaller at different box scales of r than the height of boxes in the DBC algorithm and is calculated by Equation (22); r is defined in Equation (23) as follows:

$$\mathrm{h} = \frac{\mathrm{s}}{1+2\mathrm{b}\times\delta_\mathrm{I}} \ . \tag{22}$$

$$r = {}^{s}/_{M} \ . \tag{23}$$

Where, b is a positive integer whose optimum value is equal to 3 according to [15],

9) Calculate the total number of boxes that cover the image, using Equation (24).

$$N_r = \sum n_r(i,j) \ . \tag{24}$$

10) Estimate the fractal dimension of the image from the least squares linear fit of *log* (N_r) versus *log (1/r)*.

5. EXPERIMENTAL RESULTS

Eight images taken from the Background database are shown in Fig. 3. These are randomly selected and applied to identify the effects of noise on the fractal dimensions of the images. We first calculate the fractal dimensions of images without noise by using the original DBC and the algorithms in [16], [15] and [14]. We then add Gaussian, Salt-and-Pepper and Speckle noises to the images and recalculate their fractal dimensions via the aforementioned algorithms. We consider the default values for these noises, therefore, Gaussian noise with a mean of 0 and 0.01 variance, Salt-and-Pepper noise with a density of 0.05 and Speckle noise with a mean of 0 and 0.04 variance are applied. As shown in the Table 1, the three kinds of noises exert notable effects on the fractal dimensions, which increase under the presence of the noises. This result is attributed to the noise-induced changes in image roughness. In other words, the fractal dimensions of the gray-level images substantially change with increasing variance or density of the three noises. The estimation results derived by the original DBC and the algorithms presented in [16], [15] and [14] and the effects of noise are illustrated in Figs. 4-8.

The findings indicate that these methods are non-resistant to noise, making modifications to DBC and other algorithms with no image preprocessing necessary.

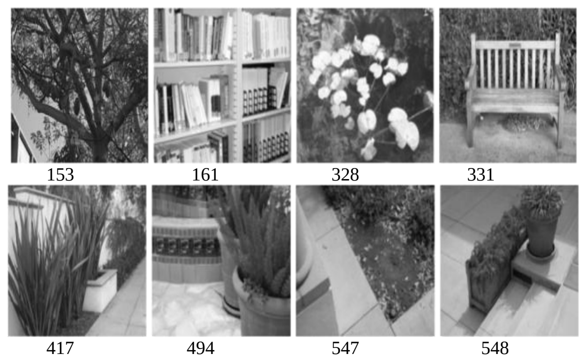

Fig. 3. Images from Background image database.

Table 1 and Fig. 4 also show the experimental results of fractal dimension calculation by the proposed method before and after the addition of the different kinds of noises. The matching of fractal dimension values on a plot indicates that the proposed method exhibits effective de-noising, desirable performance, and robustness against noise.

According to Table 1, although the fractal dimensions obtained by the proposed method are higher than those of the other methods, the fractal dimensions are in the correct interval between 2 and 3, as well as the sequence of the complexity of the images is established correctly. The reason for the high fractal dimensions in method [15] and the proposed method is to define the height of each box based on the standard deviation of the image.

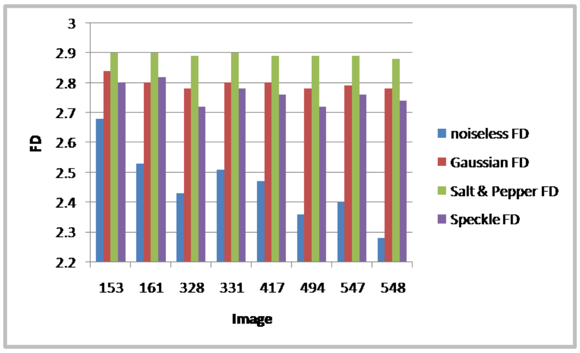

Fig. 4. Effects of noise on fractal dimension calculated by DBC.

Table 1. Results of fractal dimension estimation by DBC and similar algorithms, with and without noise.

Image	***Method***	***FD***			
		Noise less	**Gaussian Noise**	**Salt &Pepper Noise**	**Speckle Noise**
153	**DBC**	2.68	2.84	2.90	2.80
	[16]	2.67	2.84	2.92	2.79
	[15]	2.81	2.98	3.06	2.93
	[14]	2.65	2.80	2.94	2.77
	Proposed	**2.90**	**2.90**	**2.90**	**2.89**
161	**DBC**	2.53	2.80	2.90	2.82
	[16]	2.53	2.80	2.81	2.82
	[15]	2.65	2.94	3.04	2.95
	[14]	2.55	2.77	2.94	2.82
	Proposed	**2.86**	**2.85**	**2.86**	**2.85**
328	**DBC**	2.43	2.78	2.89	2.72
	[16]	2.46	2.78	2.84	2.75
	[15]	2.55	2.92	3.03	2.86
	[14]	2.47	2.77	2.96	2.73
	Proposed	**2.76**	**2.70**	**2.76**	**2.69**
331	**DBC**	2.51	2.80	2.90	2.78
	[16]	2.52	2.80	2.82	2.78
	[15]	2.63	2.94	3.04	2.92
	[14]	2.57	2.80	2.96	2.80
	Proposed	**2.81**	**2.79**	**2.81**	**2.79**
417	**DBC**	2.47	2.80	2.89	2.76
	[16]	2.50	2.79	2.83	2.77
	[15]	2.60	2.94	3.03	2.89
	[14]	2.55	2.78	2.95	2.76
	Proposed	**2.79**	**2.77**	**2.79**	**2.76**
494	**DBC**	2.36	2.78	2.89	2.72
	[16]	2.37	2.77	2.81	2.72
	[15]	2.48	2.92	3.02	2.86
	[14]	2.42	2.75	2.92	2.71
	Proposed	**2.65**	**2.56**	**2.66**	**2.56**
547	**DBC**	2.40	2.79	2.89	2.76
	[16]	2.42	2.78	2.84	2.76
	[15]	2.53	2.92	3.03	2.90
	[14]	2.47	2.76	2.94	2.75
	Proposed	**2.69**	**2.64**	**2.69**	**2.65**
548	**DBC**	2.28	2.78	2.88	2.74
	[16]	2.30	2.77	2.85	2.75
	[15]	2.39	2.91	3.02	2.88
	[14]	2.36	2.76	2.94	2.74
	Proposed	**2.57**	**2.48**	**2.58**	**2.53**

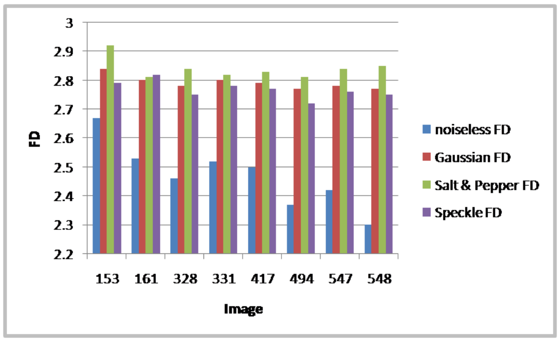

Fig. 5. Effects of noise on fractal dimension calculated by the method proposed in [16].

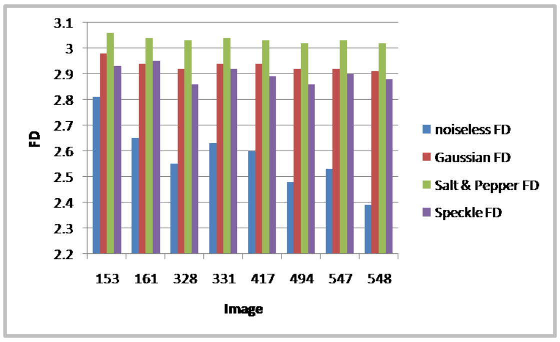

Fig. 6. Effects of noise on fractal dimension calculated by the method proposed in [15].

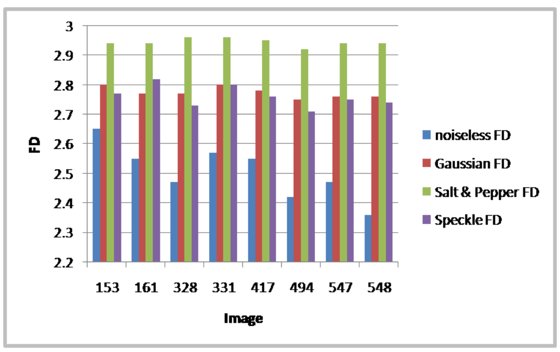

Fig. 7. Effects of noise on fractal dimension calculated by the method presented in [14].

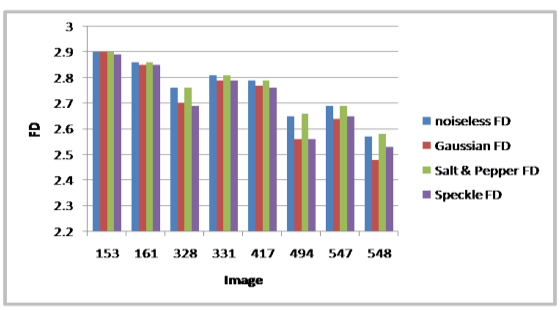

Fig. 8. Effects of noise on fractal dimension calculated by the method proposed in current work.

The variations in fractal dimensions with increasing variance or density of the Gaussian, salt-and-pepper, and speckle noises are analyzed and plotted in Fig. 9. The figure shows that the proposed method maintains robustness even under increasing noise variance or density.

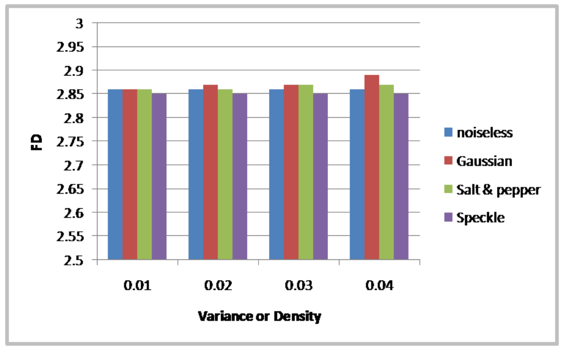

Fig. 9. Variations of fractal dimensions with increasing variance or density of noises, calculated by the method proposed in the present study.

Figs. 10-14 illustrate the log-log plots of fractal dimension estimation using different methods for "161" image. These plots show that the slop of the fitted straight-line varies in effect of noise as well. So just for that reason, fractal dimension is changed. As illustrated in Fig. 14, the slop of the fitted straight-line does not vary and so the proposed method is robust to noise.

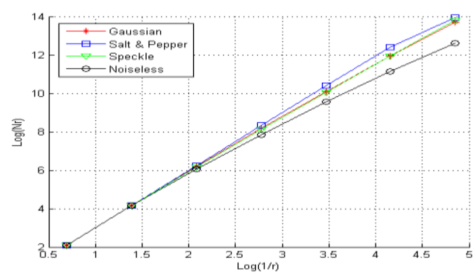

Fig. 10. The Log-Log plots of "161" image by DBC Method.

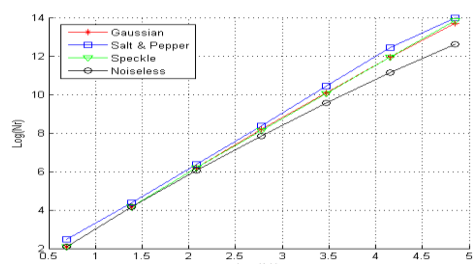

Fig. 11. The Log-Log plots of "161" image by the Method presented in [16].

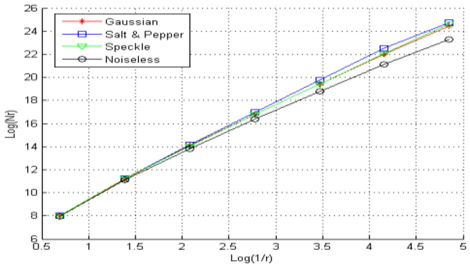

Fig. 12. The Log-Log plots of "161" image by the Method presented in [15].

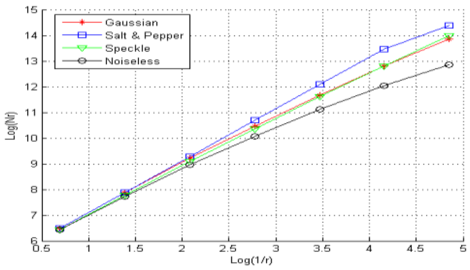

Fig. 13. The Log-Log plots of "161" image by the Method presented in [14].

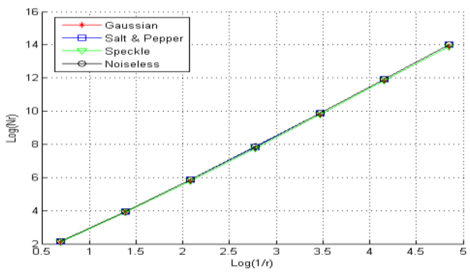

Fig. 14. The Log-Log plots of "161" image by the Method proposed in the current study.

Finally Table 2 shows the average of fractal dimension variation in effect of different noises using above methods. It is clear that the variation of fractal dimension due to different noise that is calculated by the proposed method has the lowest value compared to other algorithms.

Table 2. The comparison of the average of FD variation in effect of noise by using different methods for images of Fig. 3.

Method	Average of FD Variation (%)		
	Gaussian Noise	Salt & Pepper Noise	Speckle Noise
DBC	61.75	95	59.29
[16]	67.90	78.25	62.86
[15]	61	78.23	54.96
[14]	53.22	86.88	50.49
Proposed	**5.6**	**0.33**	**5.14**

Since the calculated fractal dimension by the estimation method is equal to the slope of the fitted straight line, the fit error *E is* used to measure the root mean-squared distance of the data pointes from the *log* (N_r) versus *log (1/r)*. The lower fit error could be obtained from the better fit. The fit error E of points *(x, y)* from their fitted straight line satisfying y=ax+b is defined by Equation (30), where *y* denotes *log* (N_r) and *x* denotes *log (1/r)*.

$$E = \frac{1}{n}\sqrt{\sum_{i=1}^{n}\frac{(ax_i+b-y_i)^2}{1+a^2}} \qquad (25)$$

The fit errors *E* computed for the images of Fig. 3 using the different methods are presented in Table 3 and compared in Fig. 15. As shown in Fig. 15, the fit errors calculated by the proposed method are less than the other methods that expresses the accuracy of the proposed method.

Table 3. The computational fit errors of FDs by using different methods for images of Fig. 3.

Image	Fit Error				
	DBC	[16]	[15]	[14]	Proposed
153	0.0253	0.0245	0.0295	0.0383	**0.0028**
161	0.0278	0.0268	0.0314	0.0353	**0.0069**
328	0.0287	0.0301	0.0318	0.0437	**0.0069**
331	0.0244	0.0232	0.0284	0.0302	**0.0042**
417	0.0205	0.0198	0.0250	0.0309	**0.0015**
494	0.0187	0.0162	0.0228	0.0197	**0.0026**
547	0.0180	0.0142	0.0230	0.0266	**0.002**
548	0.0172	0.0130	0.0217	0.0191	**0.0037**

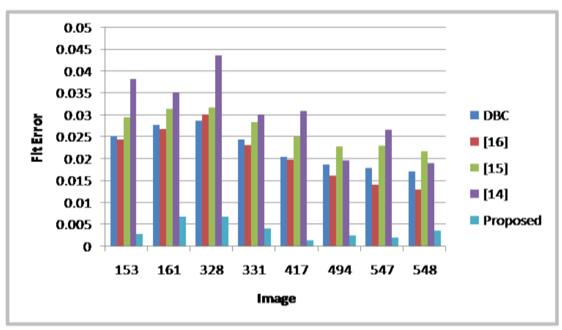

Fig. 15. The comparison of estimated FDs fit errors by using different methods.

At the end of the current study, similar to the research presented in [20], for comparison, the conventional DBC algorithm and the proposed method with median filter and mean filter preprocessing is also tested under Gaussian, salt-and-pepper, and speckle noises. In the first step, fractal dimensions of images Fig. 3, are calculated by conventional DBC algorithm and then the different noises are added. In second step, the noisy images are de-noised by median filter and mean filter, respectively, and then their fractal dimension are re-calculated by the conventional DBC algorithm. Figs. 16 and 17 show that the fractal dimensions of noisy images have improved a little by filtering. In order to approve the performance of the proposed method, also the average FD variations are calculated for noisy images of background databases using Conventional DBC method, the proposed method and filtered images by median and mean filter. Fig. 18 represents that the proposed method with minimum variations, in which no preprocessing on the captured image is conducted, effectively removes the effects of the noises. It also exhibits higher efficiency and more desirable performance than those achieved using the algorithm with filtering processing.

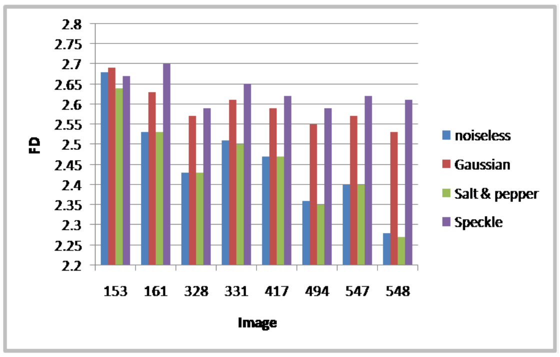

Fig. 16. Effects of noise on fractal dimensions calculated by DBC after median filtering for Background images.

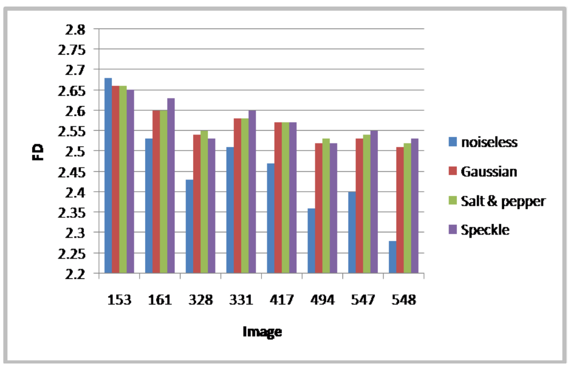

Fig. 17. Effects of noise on fractal dimensions calculated by DBC after mean filtering for Background images.

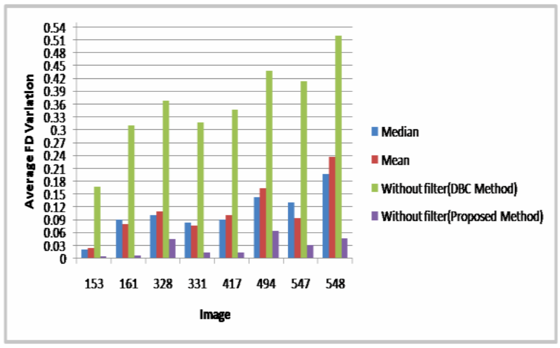

Fig. 18. Computational average FD variation of Background images.

6. CONCLUSION

The fractal dimension is an important parameter in fractal geometry, and the accuracy of its estimation guarantees the accuracy of texture image analysis. In images contaminated by noise, fractal dimensions are positively correlated with noise variance or density. Given the negative effects of noise, an image becomes rougher, and its fractal dimension becomes larger. The fractal dimension of a noisy image is larger than that of a non-noisy image. Various methods of estimating the fractal dimensions of digital images are available, but most of them are based on the original DBC, which is non robust against noise.

This study proposes a noise-robust method based on the DBC algorithm and GLCM and has tested the approach under Gaussian, salt-and-pepper, and speckle noises of different variances and densities. The experimental results show the effective de-noising performance of the proposed method.

This method contributes to the field in that it enables effective handling of the nature images, which are often exposed to noise. Solving the problems presented by noise is an important challenge in the processing of images. In such instances, noise can be removed before the images are processed, but this approach does not allow for preprocessing. The sensible strategy in this study was to analysis noisy digital images directly and without any pre-processing.

REFERENCES

[1] B.B. Mandelbrot and J. van Ness, **"Fractional Brownian Motions, Fractional Noises and Applications,"** *SIAMRev.*, Vol.10, pp. 422–437, Oct. 1968.

[2] W-L Lee and K-Sh Hsieh, **"A Robust Algorithm for the Fractal Dimension of Images and its Applications to the Classification of Natural Images and Ultrasonic Liver Images,"** *Signal Processing*, Vol. 90, pp. 1894–1904, Jun. 2010.

[3] P. Pentland, **"Fractal-Based Description of Natural Scenes,"** *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 6, pp. 661–674, Nov. 1984.

[4] T. Pant, D. Singh and T. Srivastava, **"Advanced Fractal Approach for Unsupervised Classification of SAR Images,"** *Advances in Space Research*, Vol. 45, pp. 1338-1349, Jun. 2010.

[5] X. Zhang, Y. Xu and Jackson, R.L., **"An Analysis of Generated Fractal and Measured Rough Surfaces in Regards to their Multi-Scale Structure and Fractal Dimension,"** *Tribology International*, Vol. 105, pp. 94–101, Jan. 2017.

[6] X. Zhao and X. Wang, **"Fractal Dimension Estimation of RGB Color Images Using Maximum Color Distance,"** *World Scientific Publishing Company, Fractals*, Vol. 24, pp.1-7, Aug. 2016.

[7] M. Fernández-Martínez and M.A. sánchez-Granero, **"A New Fractal Dimension for Curves based on Fractal Structures,"** *Topology and its Applications*, Vol. 203, pp.108–124, Apr. 2016.

[8] E. Spodarev, P. Straka and S. Winter, **"Estimation of Fractal Dimension and Fractal Curvatures from Digital Images,"** *Chaos, Solitons& Fractals***,** Vol. 75, pp. 134-152, Jun. 2015.

[9] C. Yinglei, et al. (2010), **"A Method of Calculating Image Fractal Dimension Based on Fractal Brownian Model,"** *in Proc. 2010 International Forum on Information Technology and Applications, Kunming, IEEE, China*, pp.19-21.

[10] H.O. Peitgen, H. Jurgensand D. Saupe, "**New Frontiers of Science,"** in *Chaos and Fractals*, 2nd ed., 2004, Springer.

[11] N. Sarkar and B.B. Chaudhuri, **"An Efficient Approach to Estimate Fractal Dimension of Textural Images,"** *Pattern Recognition*, Vol. 25, pp. 1035–1041, Sep. 1992.

[12] N. Sarkar and B.B. Chaudhuri, **"An Efficient Differential Box-counting Approach to compute Fractal Dimension of Image,"** *IEEE Transactions Systems, Man and Cybernetics*, Vol. 24, pp. 115–120, Jan. 1994.

[13] N. Sarkar and B.B. Chaudhuri, **"Multifractal and Generalized Dimensions Of Gray Tone Digital Images,"** *Signal Processing*, Vol. 42, pp.181–190, Mar. 1995.

[14] Y. Shi, et al., **"An Anti-noise Determination on Fractal Dimension for Digital Images,"** *in Proc. 2011 Advances in Intelligent and Soft Computing, Berlin, Springer-Verlag, Germany*, pp. 469-474.

[15] J. Li, Q. Du and C. Sun, **"An Improved Box-counting Method For Image Fractal Dimension Estimation,"** *Pattern Recognition*, Vol. 42, pp. 2460–2469, Nov. 2009.

[16] X.C. Jin, S.H. Ong and Jayasooriah, **"A Practical Method for Estimating Fractal Dimension,"** *Pattern Recognition Letter*, Vol. 16, pp. 457–464, May. 1995

[17] A. Eleyan and H. Demirel, **"Co-occurrence Matrix and its Statistical Features as a New Approach for Face Recognition,"** *Turkish Journal of Electrical Engineering and Computer Sciences*, Vol. 19, pp. 97-107, Dec. 2011.

[18] R.M. Haralick, K. Shanmugam and I. Dinstein, **"Textural Features for Image Classification,"** *IEEE Transactions on Systems, Man and Cybernetics***,** Vol. 3, pp. 610–621, Nov. 1973.

[19] P. Mohanaiah, P. Sathyanaray and L. Gurukumar, **"Image Texture Feature Extraction Using GLCM Approach,"** *International Journal of Scientific and Research Publications*, Vol. 3, pp. 2250-3153, May. 2013.

[20] S. R. Nayak and J. Mishra**, "On Estimation of Fractal Dimension of Noise Images,"** *Indian journal of science and Technology*, Vol. 10, pp. 1-6, May. 2017.