



## RECOGNITION OF IRREGULARLY SHAPED WORDS BY USING FRACTAL DIMENSION

*Emre ERKAN<sup>1\*</sup>, Muhammet Ali ARSERİM<sup>2</sup>, Mehmet Sıraç ÖZERDEM<sup>2</sup>*

<sup>1</sup>Batman University Vocational School, Batman, Turkey

<sup>2</sup>Dicle University Engineering Faculty Department of Electric-Electronic Engineering, Diyarbakir, Turkey

\* Corresponding author; E-mail: emre.erkana@batman.edu.tr

Received: 14 September 2018; Accepted: 17 November 2018

**Abstract**— Today, optical character recognition technology is not so advanced as to compete with human perception ability. Parameters such as scene complexity, irregular lighting conditions, skewness, blur and distortion, aspect ratios, perspective impairment, fonts, multilingual environments negatively affect the success of the optical character recognition technology. The aim of this article is to create an algorithm that can resolve irregular words whose characters' scales and rotations are modified. In the algorithm, fractal dimension tool, a fast and stable recognition method, is used. From this viewpoint it is desired to make optical character recognition technology closer to human perception. In order to analyze the algorithm, fractal dimension and image compression data of big, and small alphabetic characters in the tahoma font were recorded in the database. Then, using these characters, irregular word images were obtained. These images, were analyzed by the algorithm built in matlab program and the results were obtained.

**Key words:** Fractal Dimension, Pattern Recognition, Optical Character

### 1. Introduction

Optical Character Recognition (OCR) is the process of replacing or converting a document containing text or any text, such as handwriting, printed, or scanned document images, into an editable digital format for deeper and further processing. Today, the recognition of machine characters has largely been solved [1], but OCR technology is not as advanced as to be able to compete with the human perception ability. Effects such as scene complexity, irregular lighting conditions, skewness, blur and distortion, aspect ratios, perspective disturbance, typeface, multilingual environments negatively affect OCR technology's ability [2].

There are many studies on the fractal dimension in the literature, but studies on texts are limited. The foundation is based on fractal dimension; Finding text distortion in the image [3], detection of fonts in Arabic texts [4], detection of which text is written in alphabet [5], text detection and recognition from

mobile video images [6]. However, there is no fractal dimension-based approach that can improve recognition of rotated characters, one of the weaknesses of optical character recognition technology. The aim of this study is to develop an algorithm that can enable the recognition of words that are rotated to their characters. For this purpose, two important applications of the fractal dimension, a powerful and rapid recognition tool; box counting method and image compression are the basis of the designed algorithm [7,8].

## 2. Phases of the Algorithm

The first phase of the algorithm begins with a character determination process. This determination process is specifically designed for the designed algorithm. The next step is the image compression phase. In this way, images becomes independent from scale and rotation. The third phase is the calculation of the fractal dimension. The final stage is the interpretation of the data and the realization of the character recognition process [7].

Each character is passed through the above mentioned stages in order to analyze the word that forms the image.

This algorithm was developed in the MATLAB program, which was tested for irregular words consisting of large and small alphabetic characters defined in the Tahoma font.

### 2.1. Determination of Characters

Characterization is specially designed for this algorithm. An example of a Word, formed irregularly is given in Fig.1.

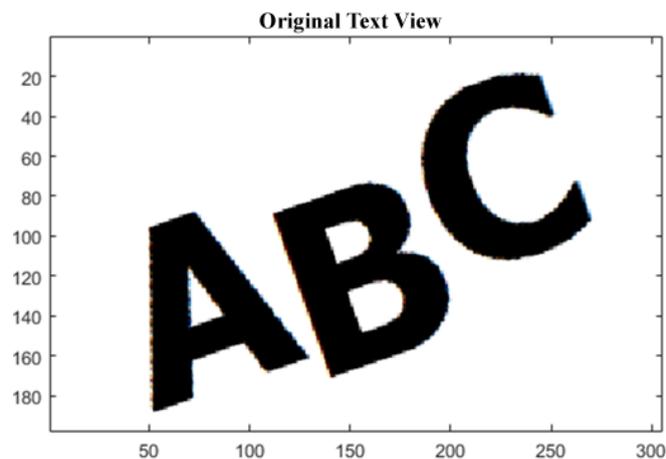
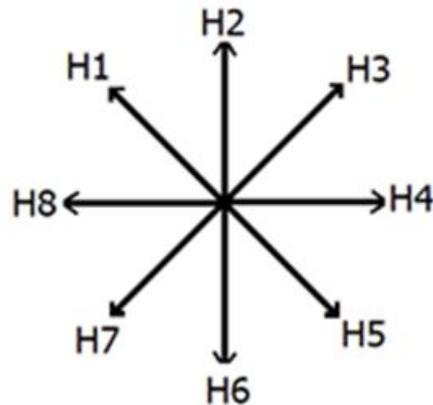


Fig. 1. Irregular vocabulary example

In this example, the image is encoded in binary form. For characterization, the image columns are scanned from top to bottom and from left to right, respectively. The first black pixel detected is used as the starting pixel of the first character boundary to be tested. To obtain the boundaries of the character whose starting point is determined, the boundary pixels of the character in the clockwise direction are found. Thus, the boundary of the character is extracted.



**Fig. 2.** Directions used when determining the boundary for a character

The directions in Fig. 2 are used when determining the boundaries of the character. Some important situations are need to be known for the boundary detection algorithm. When the starting pixel is detected, the first direction of motion is H6. The determination of the next boundary pixel is depended on the direction of the previous movement.

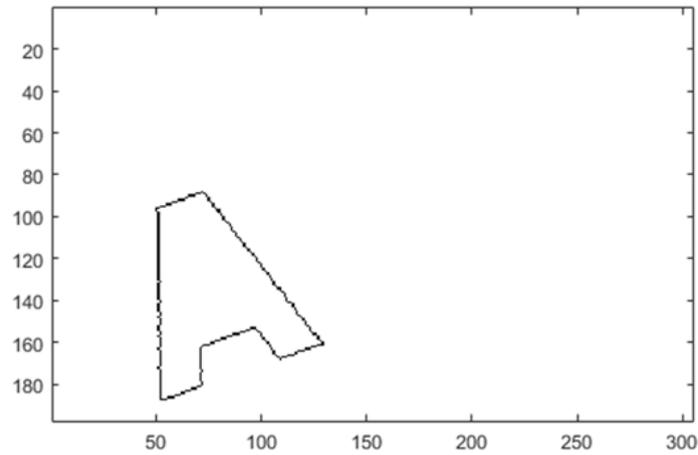
For example, find the next boundary pixel for the boundary pixel with the final direction of motion H6;

The direction of movement can not be H2, because the aim is to determine the limit of the character in clockwise direction. Since H2 means that the movement is returned to the direction where it comes from. For the clockwise determination of the border, the pixels of the H3-H4-H5-H6-H7-H8-H1 must be tested sequentially. Always if the corresponding pixel is within the boundary of the character, then the direction of the last movement will be through this pixel. For example, when the directions are tested if the pixel in the H5 direction is found to be black, the final direction of motion is determined as H5. This process continues until the starting pixel is the same as the pixel specified after the last transaction. Thus a closed boundary of the character is obtained. Tab. 1. shows the directions in which the pixels should be tested, respectively, according to the direction of the final motion.

**Tab.1.** Pixel directions to be tested by last motion direction

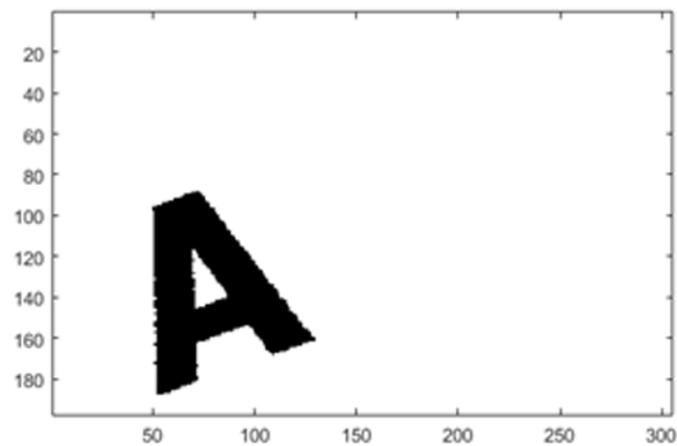
Last Movement Direction	Pixel Aspects Needed to Test Sequentially
H1	H6-H7-H8-H1-H2-H3-H4
H2	H7-H8-H1-H2-H3-H4-H5
H3	H8-H1-H2-H3-H4-H5-H6
H4	H1-H2-H3-H4-H5-H6-H7
H5	H2-H3-H4-H5-H6-H7-H8
H6	H3-H4-H5-H6-H7-H8-H1
H7	H4-H5-H6-H7-H8-H1-H2
H8	H5-H6-H7-H8-H1-H2-H3

When the algorithm is executed according to the rules in table 1, the boundary in Fig. 3 is obtained.

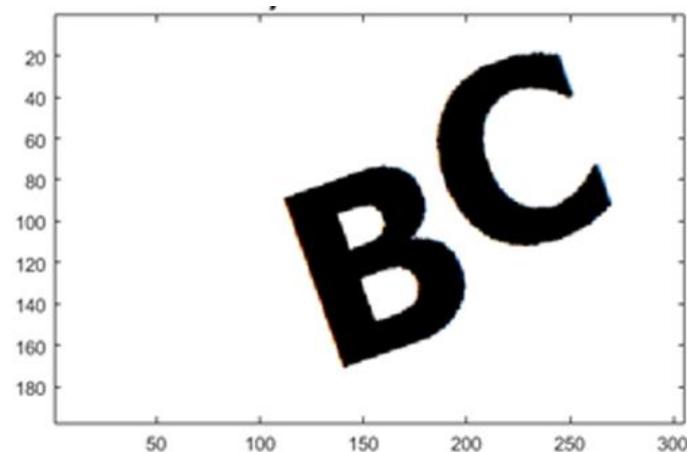


**Fig. 3.** Determining the boundaries of the first character

The boundary of the first character is shown in the Fig. 3. All the data located in the bounds of the specified character is taken from the original image as in Fig. 4. Thus, the defined character is separated from the original image. For the next character recognition operation, the previously specified character is subtracted from the original image and this is shown in Fig. 5.



**Fig. 4.** Retrieving the first character data from the original image

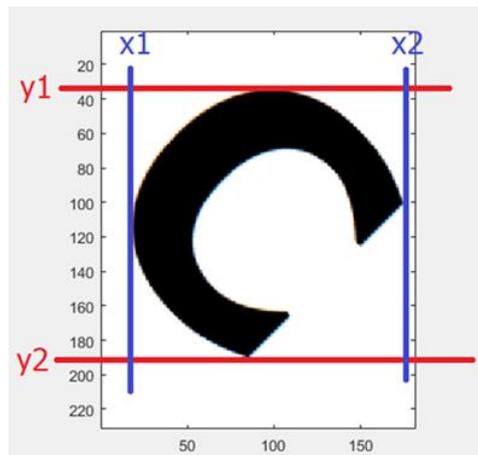


**Fig. 5.** Previously specified character is removed from the original image for the next step

## 2.2. Image Compression

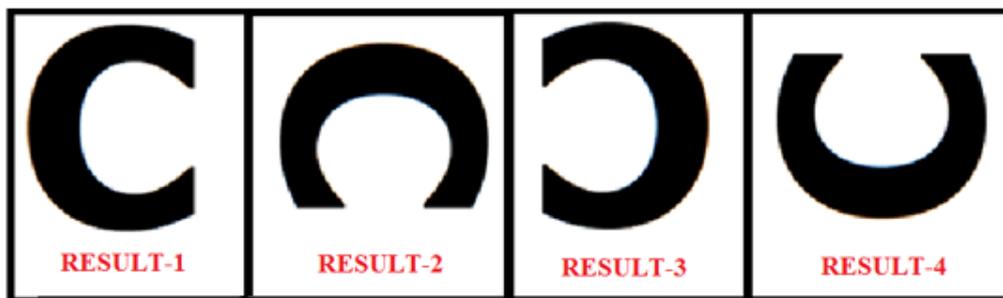
It is possible to make the recognition process independent from the scale and the rotation by using picture compression, one of the image processing applications of fractals [9].

In order to make the character independent from the rotation, pre-processing should be done. First, the character specified from the original image is subjected to rotation from 1 to 90 degrees. After each 1 degree rotation, the area associated with the character as in Fig. 6 is calculated as  $(x_2-x_1) * (y_2-y_1)$  [7].



**Fig.6.** Rotated character

The smallest angle value is obtained as a result of these transformations and the character is rotated by the determined angle value. As a result of this process, 4 different results can be obtained. These results are shown in Fig. 7.



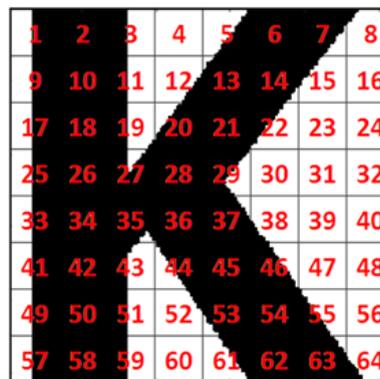
**Fig. 7.** Cases that may occur after rotation transactions applied to the character

A large number of results means that a large number of transactions are being made in the recognition phase. the number of results can be reduced by half with a simple process; If the character's width is greater than the height, the character is repeated with a rotation of 90 degrees, otherwise it is left in its current position. In this way, the number of results is reduced to two, which is shown in Fig. 8 [7].



**Fig. 8.** Reduction of results

In the final stage of image compression, the domain of the character is found; the result obtained by applying rotation is placed as in Figure 9, which is equal to the size of a 256x256 Pixel white image, and the width is centered on it. A 256x256-Pixel white image can be larger or smaller. As this field grows, the domain of the character is better calculated. However, the growth of this area means the increase in the transaction load.



**Fig. 9.** Finding the character domain

Then the resulted image is divided into equavelent sections consisting of 8x8 64 squares and enumerated as in Fig. 9. Section overlapping with character is assigned "1" whereas "0" is assigned to non-overlapping parts. Thus, the 64 bit character domain data is generated and this data is stored in the database for using in the recognition phase.

### 2.3. Finding a Fractal Dimension with Box Counting Method

The box counting method can be used not only to find the fractal size of the fractal, but also to detect fractal size of non-fractal objects and characters [10].

The image is usually divided into parcels of 32x32, 16x16, 8x8, 4x4 or 2x2 pixels to find the fractal dimension of an object or a character. The smaller the size of the section, the more accurate the calculated fractal dimension will be [11,12,13].

$$Fractal\ Dimension = \frac{\log(N)}{\log\left(\frac{1}{s}\right)} \quad (1)$$

where N is the number of sections affected by the image, whereas 1/s represents the inverse of the parcel fields.

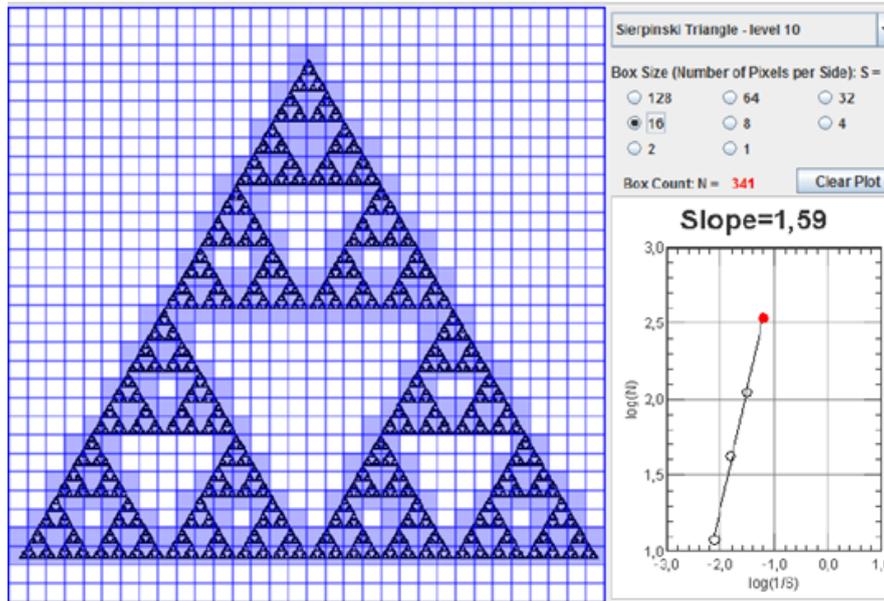


Fig. 10. 16x16 Pixel parcelled image

In Fig. 10 and Fig. 11, the fractal dimension is calculated for different section sizes. the fractal dimension is equal to the slope in the chart as shown in the shapes,. The bigger the slope, the more fractal the object is. The complexity of the image increases as the fractness increases.

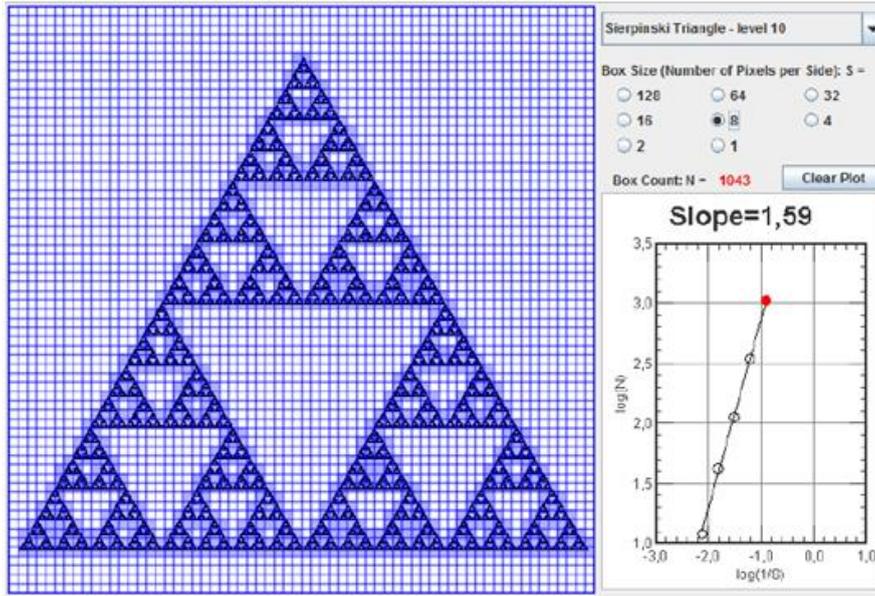


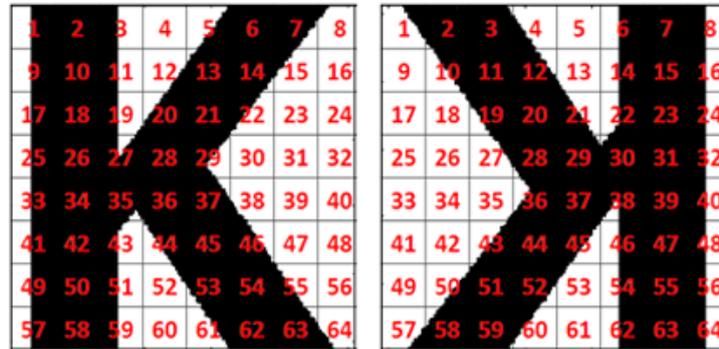
Fig. 11. 8x8 Pixel parcelled image

When it is desired to define a character or object, decision made just by looking at the fractal size or only the domain is not the right approach. For this reason, for more accurate results, the 64-bit domain data and the character's fractal dimension are stored in the database when defining characters.

## 2.4. Character Recognition

In the first step of the algorithm, the character is determined and the fractal dimension of the detected character is calculated. Then, the characters that have similar fractal dimensions are compared with the

both normal, and 180 degrees rotated domains, shown as in Fig. 12, of the character, found in recognition process.



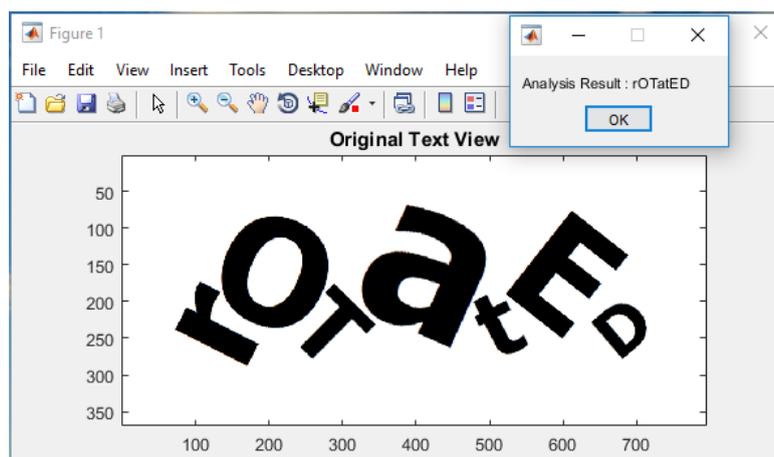
**Fig. 12.** Domains for the character

As a result of this comparison the character is assigned to the domain of a character with respect to which the character is most similar, with a condition that it does not exceed a specific error margin. The same actions are repeated for all characters that make up the image. By combining the characters that are resolved sequentially, the irregular word is solved.

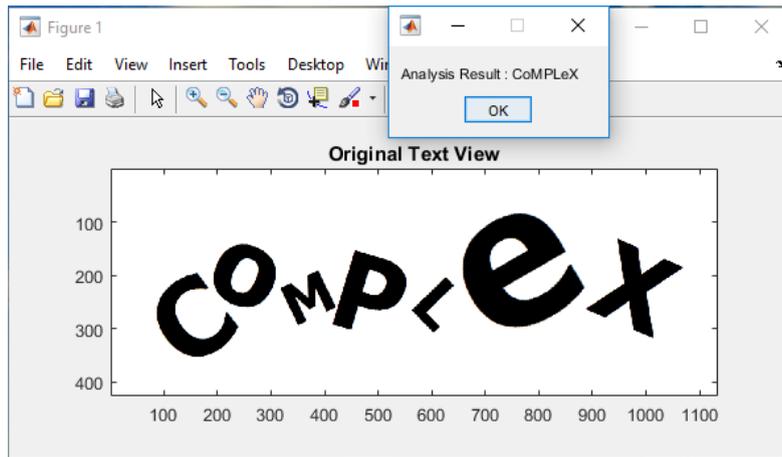
### 3. Conclusions

In order to test the algorithm, sample irregular words were used in Tahoma font. The generated images can be seen in Fig. 13, 14, 15, 16. The characters that make up the words in different scales, different rotations, case sensitive, various locations were tested. No interference was performed to eliminate noise in the algorithm and noise-free images were used for the test.

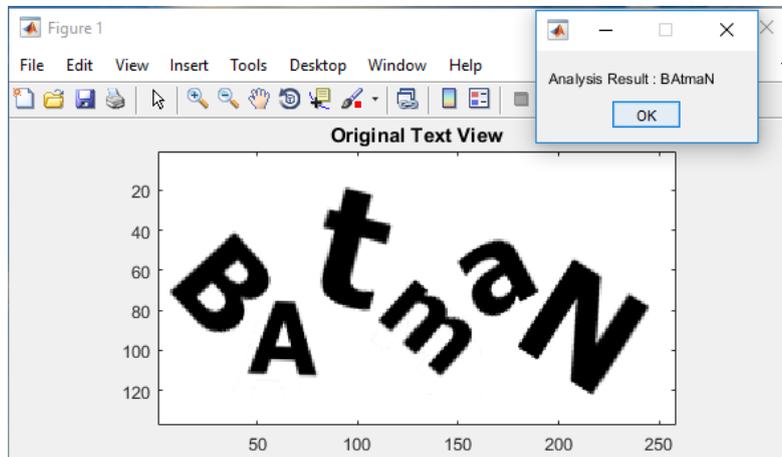
As a result of the tests, it is observed that this algorithm based on fractal size has achieved successful results. Thanks to this algorithm, optical character recognition technology can be made more stable against character rotation. Thus, optical character recognition technology is projected to be one step closer to human perception.



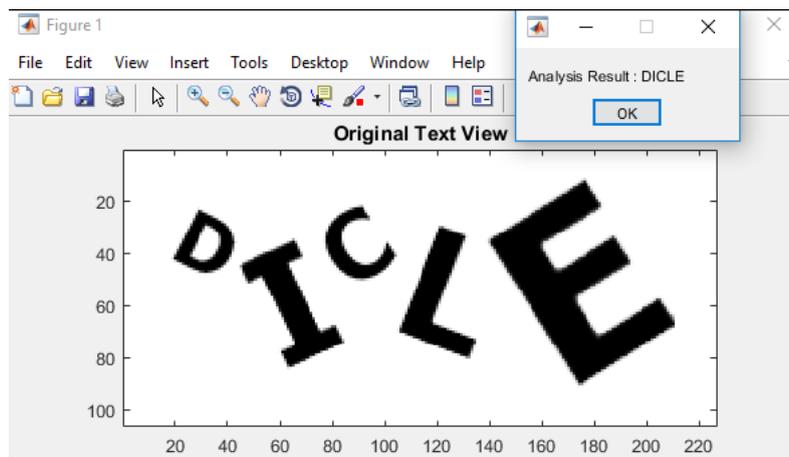
**Fig. 13.** Resolving of the irregular 'rOTatED' word



**Fig. 14.** Resolving of the irregular 'CoMPLeX' word



**Fig. 15.** Resolving of the irregular 'BATmaN' word



**Fig. 16.** Resolving of the irregular 'DICLE' word



## References

- [1] Hamad K.A., Kaya M., A Detailed Analysis of Optical Character Recognition Technology, *International Journal of Applied Mathematics, Electronics and Computers*, 4 (Special Issue) (2016), pp.244-249, ISSN: 2147-82282147.
- [2] Vamvakas G., Gatos B., Perantonis S.J., Handwritten Character Recognition through Two-Stage Foreground Sub-Sampling, *Pattern Recognition*, 8 (2010), pp.2807-2816, DOI No: 10.1016/j.patcog.2010.02.018.
- [3] Yu C.L., Tang Y.Y., Suen C.Y., Document Skew Detection Based on the Fractal and Least Squares Method, *Proceedings of 3rd International Conference on Document Analysis and Recognition*, Montreal, Quebec, Canada, 1995, Vol. 1, pp. 1149-1152.
- [4] Moussa S.B., Zahour A., Benabdelhafid A., Alimi A.M., New Features Using Fractal Multi-Dimensions for Generalized Arabic Font Recognition, *Pattern Recognition Letters*, 5 (2010), pp.361-371, DOI No: 10.1016/j.patrec.2009.10.015.
- [5] Moussa S.B., Zahour A., Benabdelhafid A., Alimi A.M., Fractal-Based System for Arabic/Latin, Printed/Handwritten Script Identification, *2008 19th International Conference on Pattern Recognition*, Tampa, FL, USA, 2008, Vol. 1, pp. 978-981, DOI No: 10.1109/ICPR.2008.4761838.
- [6] Shivakumara P., Wu L., Lu T., Tan C.L., Blumenstein M., Anami B.S., Fractals Based Multi-Oriented Text Detection System for Recognition in Mobile Video Images, *Pattern Recognition*, 68 (2017), pp.158-174, DOI No: 10.1016/j.patcog.2017.03.018.
- [7] G. Neil, K. M. Curtis, Shape Recognition Using Fractal Geometry, *Pattern Recognition*, Vol. 30, No. 12 (1997), pp. 1957-1969.
- [8] Les T., Kruk M., Osowski S., Objects Classification Using Fractal Dimension And Shape Based On Leaves Classification, *ResearchGate*, June 2013.
- [9] D. C. Popescu and H. Yan, Fractal-Based Method for Colour Image Compression, *J. Electronic Imaging*, Vol.4(1) (1995), pp.23-30.
- [10] Fractal Dimension- Chapter 2, <http://fractalfoundation.org/OFC/OFC-10-1.html> [Access February 7, 2018].
- [11] M. Sonaka, V. Hlavac and R. Boyle, *Image Processing, Analysis and Machine Vision*, Chapman and Hall Computing, London, 1993.
- [12] M. Cusenza, *Fractal Analysis of the EEG and Clinical Applications*, Università Degli Studi Di Trieste, Trieste, Italy, 2012.
- [13] Buczkowski S., Kyriacos S., Nekka F., Cartilier L., The Modified Box-Counting Method: Analysis of Some Characteristic Parameters, *Pattern Recognition*, 4 (1998), pp.411-418, DOI No: 10.1016/S0031-3203(97)00054-X