Sar Image Segmentation Using Modified Bacterial Forging Optimization Algorithm

Abhijit Kaur¹, Amandeep Singh Bhandari²

Computer Science and Engineering Sri Guru Granth sahib World University¹
Assistant professor, Department of Electronics and Communication, Punjabi University Patiala²

Abstract

This paper provides a detailed study of the image segmentation on SAR and captcha images. A fast segmentation method is planned for SAR and captcha images. The technique regards threshold estimation as a search process and employs Modified BFO algorithm to optimize it. In order to provide Modified BFO algorithm with an efficient fitness function, the concept of grey number in Grey theory is integrated, maximum conditional entropy to get an improved two-dimensional grey entropy. In essence, the fast segmentation speed of our method owes to Modified BFO algorithm, which has an outstanding convergence performance. On the other hand, the segmentation quality of our method is benefit from the improved two-dimensional grey entropy, for the fact that noise almost completely disappears. Experimental results indicate that our method is superior to GA based, AFS based and ABC based methods in terms of segmentation accuracy, segmentation time, and thresholding.

1. Introduction

Image segmentation denotes to the separation of image into a set of areas that cover it. The goal in dissimilar tasks is for the areas to signify expressive are as of the image, such as the crops, urban zones, and forests of a satellite image. In other examination tasks, the regions might be sets of border pixels grouped into such constructions as line segments and circular arc sections in images of 3D industrial substances. Regions may also be well-defined as collections of pixels having both a border and a particular shape such as a circle or ellipse or polygon. When the interesting areas do not cover the whole image, we can still talk around segmentation, into foreground areas of interest and background areas to be overlooked [1] [2].

The prime objective of segmentation is to dissociate the image into portions for further examination. In humble cases, the environment might be well sufficient controlled so that the segmentation process reliably excerpts only the parts that need to be examined further. The segmentation is dependable, provided that the person's clothing or room background does not have the similar color components as a human face. In complex cases, such as removing a complete road network from a greyscale aerial image [3].

1.2 Thresholding

1.2.1 Foundation

Supposing that the gray-level histogram agrees to an image \( f(x, y) \) composed of light substances on a dark background, in such a way that object and background pixels have gray levels gathered into two leading modes. One clear way to excerpt the substances from the background is to select a threshold \( T \) that splits these styles. Then any point \((x, y)\) for which \( f(x, y) > T \) is called an object point; then, the point is called a contextual point. Three dominant modes characterize the image histogram (for specimen two types of light objects on a dark background). Here, multilevel thresholding categorises a point \((x, y)\) as fitting to one object class if \( T_1 < (x, y) \leq T_2 \), to the other object class if \( f(x, y) > T_2 \), and to the background if \( f(x, y) \leq T_1 \). In general. Segmentation glitches needful multiple thresholds are best solved using region growing approaches [4].

To see in what way this works, recall that in MATLAB, a process on a single number, when practical to a matrix, is understood as being practical concurrently to all rudiments of the matrix. The command \( X > T \) Will thus reappearance 1 (for true) for all those pixels for which the grey standards are greater than \( T \), and 0 (for false) for all those pixels for which the grey values are less than or equivalent to \( T \). We thus end up with a matrix of 0's and 1's, which can be watched as a binary image [4] [5] [6].
1.3 IMPLEMENTATION STARTS
1.3.1 Detection of Discontinuities

For detecting the three basic types of gray-level breaks in a digital image: points, lines, and edges, this process involves computing the sum of crops of the coefficients with the gray levels enclosed in the region encompassed by the mask [7].

1.3.2 Point Detection

The detection of remote points in an image is frank in principle. We say that a point has been noticed at the site on which the mask is centred if [11]

\[ |R| \geq T \]

Where \( T \) is a non-negative threshold. Essentially, this preparation measures the weighted changes among the center point and its neighbours. The idea is that an inaccessible point (a point whose gray level is meaningfully dissimilar from its [5] background.

1.3.3 Line Detection

The next level of difficulty is line detection. If the first mask were enthused around an image, it would reply more strongly to lines (one pixel thick) concerned with horizontally. With a continuous background, the maximum reply would result when the line approved finished the middle row of the mask [8] [12].

1.3.4 Edge Linking and Boundary Detection

This method for applying first- and second-order digital offshoots for the detection of edges in an image.

1.3.5 Thresholding

A greyscale image is twisted into a binary (black and white) image by first selecting a grey level \( T \) in the original image, [1] and then turning every pixel black or white rendering to whether it’s grey

Value is greater than or less than \( T \)

A pixel becomes

Thresholding is a vital part of image segmentation, where we wish to isolate substances from the contextual. It is also a significant constituent of robot vision. The subsequent image can then be further treated to find the number, or average size of the grains [9].

In this trial, for Modified BFO algorithm, the population size is 10, the fixed maximum number of iterations is 10, and the limit times for desertion is 10, the lower and upper bounds are 0 and 255 respectively [10].

1.4 Result

1.4.1 Case 1

![Figure 1: Original Image](Image)

![Figure 2: Segmented Image](Image)

We take an image and calculate the threshold value

79, Time elapsed 7.354220 seconds, and image Fitness is \( 1.0 \times 10^{03} \times 1.4125 \).

1.4.2 CASE 2

![Figure 3: Original Image](Image)

![Figure 4: Segmented Image](Image)

We take an image and calculate the threshold value

53, Time elapsed 7.472532 seconds, and Image Fitness is \( 1.0 \times 10^{03} \times 1.4268 \).

1.4.3 CASE 3

![Figure 5: Original Image](Image)

![Figure 6: Segmented Image](Image)

We take an image and calculate the threshold value

142, Time elapsed 7.810519 seconds, and Image Fitness is \( 1.0 \times 10^{03} \times 3.5259 \).
1.4.4 CASE 4

![Original Image](image1.png) ![Segmented Image](image2.png)

Figure 7: Original Image  Figure 8: Segmented Image

We take an image and calculate the threshold value 165. Time elapsed 7.545454 seconds, and Image Fitness is $1.0 \times 10^{3} \times 3.8507$.

1.4.5 CASE 5

![Original Image](image3.png) ![Segmented Image](image4.png)

Figure 9: Original Image  Figure 10: Segmented Image

We take an image and calculate the threshold value 154. Time elapsed 7.677205 seconds, and Image Fitness is $1.0 \times 10^{3} \times 2.2579$.

1.4.6 CASE 6

![Original Image](image5.png) ![Segmented Image](image6.png)

Figure 11: Original Image  Figure 12: Segmented Image

We take an image and calculate the threshold value 154. Time elapsed 7.677205 seconds, and Image Fitness is $1.0 \times 10^{3} \times 2.2579$.

1.4.7 CASE 7

![Original Image](image7.png) ![Segmented Image](image8.png)

Figure 13: Original Image  Figure 14: Segmented Image

We take image and calculate the threshold value, Time elapsed, and fitness of an image, and compare the results with the previous work taken in the Table below show the comparison of ant colony, Genetic, Artificial Fish Swarm algorithms. Image Fitness is $1.0 \times 10^{3} \times 2.5686$

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Fitness Function</th>
<th>Threshold</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MBFO</td>
<td>Improved two-dimensional grey entropy</td>
<td>167</td>
<td>6.1639</td>
</tr>
<tr>
<td>ACS</td>
<td>Improved two-dimensional grey entropy</td>
<td>205</td>
<td>5.821</td>
</tr>
<tr>
<td>GA</td>
<td>Two-dimensional entropy</td>
<td>207</td>
<td>14.391</td>
</tr>
<tr>
<td>GA</td>
<td>Two-dimensional grey entropy</td>
<td>206</td>
<td>17.640</td>
</tr>
<tr>
<td>AFS</td>
<td>Two-dimensional Otsu</td>
<td>187</td>
<td>12.015</td>
</tr>
</tbody>
</table>

1.4.8 CASE 8

![Original Image](image9.png) ![Segmented Image](image10.png)

Figure 15: Original Image  Figure 16: Segmented Image

We take image and calculate the threshold value, Time elapsed, and fitness of an image, and compare the results with the previous work taken in the Table below show the comparison of ant colony, Genetic, Artificial Fish Swarm algorithms. Image Fitness is $1.0 \times 10^{3} \times 2.4109$. 

Table 1: Threshold and Time Comparison over various algorithms
Table 2: Threshold and Time Comparison over various algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Fitness Function</th>
<th>Threshold</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MBFO</td>
<td>Improved two-dimensional grey entropy</td>
<td>165</td>
<td>5.989</td>
</tr>
<tr>
<td>ACS</td>
<td>Improved two-dimensional grey entropy</td>
<td>204</td>
<td>6.669</td>
</tr>
<tr>
<td>GA</td>
<td>Two-dimensional entropy</td>
<td>163</td>
<td>15.35</td>
</tr>
<tr>
<td>GA</td>
<td>Two-dimensional grey entropy</td>
<td>207</td>
<td>19.15</td>
</tr>
<tr>
<td>AFS</td>
<td>Two-dimensional Otsu</td>
<td>162</td>
<td>12.54</td>
</tr>
</tbody>
</table>

1.4.9 CASE 9

Figure 17: Original Image
Figure 18: Segmented Image

We take image and calculate the threshold value, Time elapsed, and fitness of an image, and compare the results with the previous work taken in the Table below show the comparison of ant colony, Genetic, Artificial Fish Swarm algorithms. Image Fitness is 881.162

Table 3: Threshold and Time Comparison over various algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Fitness Function</th>
<th>Threshold</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MBFO</td>
<td>Improved two-dimensional grey entropy</td>
<td>61</td>
<td>5.965</td>
</tr>
<tr>
<td>ACS</td>
<td>Improved two-dimensional grey entropy</td>
<td>95</td>
<td>4.835</td>
</tr>
<tr>
<td>GA</td>
<td>Two-dimensional entropy</td>
<td>131</td>
<td>13.460</td>
</tr>
<tr>
<td>GA</td>
<td>Two-dimensional grey entropy</td>
<td>94</td>
<td>17.740</td>
</tr>
<tr>
<td>AFS</td>
<td>Two-dimensional Otsu</td>
<td>62</td>
<td>6.441</td>
</tr>
</tbody>
</table>

1.5 COMPARISON OF NATURE-INSPIRED ALGORITHMS

The above experimental results indicate that our method outperforms the methods in [GA, ABS, and AFS]. To compare the convergence performance of Modified BFO algorithm, GA, and AFS algorithm, this group of experiments run with the same settings, comprising the fitness functions (improved two-dimensional grey entropy), the maximal iterations (10), the population size (50) with the same initial population distribution, over the 10 runs. The traces of fitness and thresholds are given. Moreover, ABC algorithm converges very quickly, especially at initial part. Sometimes the threshold does not change for several iterations, but dramatically changes at some iterations.

1.6 COMPARISONS OF SEGMENTATION TIME

To compare the segmenting time spent in Figures, some experimental results are listed in Tables above.

The comparative results in Tables indicate that our method is significantly faster than the other three methods in. The segmenting time of the methods tested here is ordered as our method < the method in ACS < the method in AFS < the method in GA < the method in GA.
On the other hand, significantly our algorithm better than the thresholds of GA and AFS based methods. Particularly, Tables show that our method is robust to noise pollution for the fact that the segmentation threshold (167) of the optical image polluted by synthetic noise is so close to the segmentation threshold (165) of the noise-free optical image and the segmentation threshold in third case is (61).

1.7 CONCLUSION

We proposed a fast segmentation method on SAR and captcha images. The technique regards threshold estimation as a search process and employs Modified BFO algorithm to optimize it. In order to provide Modified BFO algorithm with an efficient fitness function, we integrate the concept of grey number in Grey theory, maximum conditional entropy to get an improved two-dimensional grey entropy. In essence, the fast segmentation speed of our method owes to Modified BFO algorithm, which has an outstanding convergence performance. On the other hand, the segmentation quality of our method is benefit from the improved two-dimensional grey entropy, for the fact that noise almost completely disappears. Experimental results indicate that our method is superior to GA based, AFS based and ABC based methods in terms of segmentation accuracy, segmentation time, and Thresholding.

References


