Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 9, No. 4, 2021, 439-449.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

Object Segmentation using Local Histograms, Invasive Weed Optimization Algorithm and Texture Analysis

Somayye Bayatpour and Seyed Mohammad Hossein Hasheminejad*

Department of Computer Engineering, Faculty of Engineering, Alzahra University, Tehran, Iran.

Article Info

Article History:
Received 27 October 2020
Revised 07 May 2021
Accepted 29 June 2021

DOI:10.22044/JADM.2021.10200.2158

Keywords:
Object Segmentation, Local Threshold, Histogram, Invasive Weed Optimization, Texture Analysis.

**Corresponding author: SMH.Hasheminejad@alzahra.ac.ir (S.M.H.Hasheminejad).*

Abstract

Most of the methods proposed for segmenting image objects are the supervised methods which are costly due to their requirement for large amounts of labeled data. However, in this article, we present a method for segmenting objects based on a meta-heuristic optimization that does not require any training data. This procedure consists of the two main stages of edge detection and texture analysis. In the edge detection stage, we utilize invasive weed optimization and local thresholding. The edge detection methods that are based on the local histograms are efficient methods but it is very difficult to determine the desired parameters manually. In addition, these parameters must be selected specifically for each image. In this paper, a method is presented for the automatic determination of these parameters using an evolutionary algorithm. The evaluation of this method demonstrates its high performance on natural images.

1. Introduction

Various methods have been proposed for segmenting objects in images. Most of these methods are supervised, which require collecting a large training data, making them costly and time-consuming. However, the methods based on evolutionary algorithms can make the best possible decision by choosing the best response from the population of problem solution space. They work without any requirement for training data. The edge detection methods that are based on the local histograms are efficient methods for image segmentation. Nevertheless, it is very difficult to select the optimal parameters of local thresholds manually. In addition, these parameters must be selected specifically for each image. We dynamically determined these parameters for each image using the invasive weed algorithm.

Our proposed method consists of the two stages of detection of edges and texture analysis. The edge detection is performed by an invasive weed algorithm. In each iteration, the algorithm produces new solutions according to the fitness of the previous generation's plants and at the end; the fitness value determines the best parameters.

We found that a good smoothing could improve the result of the segmentation muchly. Thus we employed the Gaussian smoothing filter whose parameters could be set and optimized for each given image. Then we optimized them using an evolutionary algorithm in order to get the best possible smoothing. In addition, we utilized invasive weed optimization (IWO) for optimization of the parameters in local thresholding. We set all mentioned parameters for both the fine and coarse edges specifically. We made two executions of edge detection for each image of dataset, and then combined the results in order to obtain a more accurate output. In the IWO execution time, the detected edge points are evaluated according to their similarity to the outer edges of the objects. Afterward, we removed the false extra edges using the texture analysis including the gradient weight transform, frequency analysis, Otsu threshold utilizing, and gradient filtering.

2. Previous Works

So far, a lot of algorithms have been developed for object segmentation and edge detection in the images. In the recent years, the evolutionary computing has been a popular approach applied in this area. Rahnamayan *et al*. [9] have provided a method for segmenting the objects using the genetic algorithm consisting of the two stages of object locating using mathematical morphology optimization and segmentation by image processing. In the optimization stage, the order of execution of morphological operators and the number of its repetitions are determined via the genetic algorithm. Ripon *et al*. [10] have used a multi-objective meta-heuristic algorithm and minimum spanning tree for segmentation. Kirchmaier *et al*. [11] have used the particle swarm algorithm. In this method, N intelligent agents traverse image windows and based on the prior knowledge, they find their final location in the image. Then they segment the image using the information of that location as well as the location of other agents. Zhang *et al*. [12] have used the Bayesian Genetic Programming for image edge detection. Tao *et al*. [13] have employed the ant colony algorithm and fuzzy entropy for segmentation. Dorrani et. al. [36] proposed an edge detection algorithm for noisy images that uses ant colony optimization. In their method, the ant's movements depend on the discrepancy of the intensity values.

A series of papers have applied evolutionary algorithms to implement multi-level-thresholding segmentation: XU *et al*. [14] have used a combination of the Dragonfly Memetic and Differential Evolution (DE) algorithms for multi-threshold segmentation, known as the improved dragonfly. The new method has no possibility to fall into the local extremes, and it uses the Otsu method to find the answers. Zhao *et al*. [15] have presented a multi-level-thresholding segmentation method with PSO algorithm and kullback_leibler divergence. In this method, the K_L divergence values between the image and its segmentation are obtained, and the sum of these values constitutes the fitness value. Banimelhem [16] has presented a multi-threshold segmentation method that uses the genetic algorithm in order to find the thresholds. The chromosomes are n * L bit vectors in it; L is the number of gray levels and n denotes the number of thresholds. The fitness is obtained by dividing the within-class variance into the between-class variance. Zhang [17] has provided a method for finding the thresholds in multi-threshold segmentation employing the ABC algorithm. The fitness of responses is determined by calculating the T_sallis entropy.

Each histogram can be modeled as a combination of the Gaussian density functions that are related to the image classes [8]. Lai *et al*. [18] have used the particle swarm algorithm in order to find the global threshold. In this method, each particle is represented by the vector containing parameters: the probability of membership in a class, the mean and the standard deviation of that class density. The PSO algorithm achieves the optimal response by minimizing the probability of wrongly assigning each pixel to classes. The method has been improved in [19]. This algorithm is called Adaptive PSO, and gives a weight to each particle based on the performance.

The Otsu method works well for the two-class segmentation. Otsu has generalized his method for a multi-level thresholding. However, as its complexity increases exponentially by the number of thresholds, this method is not applicable to real problems. Yin *et al*. [20] have provided an iterative method for finding the thresholds. In [21], the above method has been used in the initialization step of the ABC algorithm. Then the thresholds change with a certain coefficient, with the responses being selected in order to maximize the intra-class variance. Musavirad [22] has used the Differential Evolution (DE) in order to find the thresholds. Its fitness is the minimization criterion of Cross Entropy (CE) between the original image and the threshold image. Kanungo [23] has introduced a method for global thresholding with genetic algorithm, where the primary population members converge by the crowding method. The fitness function is the image histogram and the minimum value of the function is the response. Yang [24] has utilized the firefly algorithm in order to find the segmentation thresholds. The responses are given K_means as the primary seeds and it segments the image. This method eliminates the possibility of falling into the local extremes. In [32], an image segmentation method has been proposed, which uses a novel evolutionary computing algorithm. Their evolutionary method is a new model based on the behavior of the human tribes for separating their zones in one neighborhood. Each tribe represents an image segment, and each layer of method connects similar segments to each other to form a bigger segment. In [33], Arsay has used a standard firework algorithm, which is one of the swarm intelligence methods. It is exploited to optimize a K-means segmentation method. In [34], Ripon has presented a multi-objective segmentation that optimizes the three objectives of overall deviation, edge value, and connectivity measure simultaneously using the minimum spanning tree and an evolutionary algorithm (MOEA) to produce the final set of Pareto-optimal segmented image. In [35], Widynski has proposed a novel contour detection algorithm, which tracks small pieces of edges called edgelets at two scales. The edgelets embed the semi-local information, and is the basic element of the proposed recursive Bayesian modeling. They used the color and gradient information via the local, textural, oriented, and profile gradient-based features.

3. Invasive Weed Algorithm

This algorithm was developed in 2006 by Mehrabian and Lucas [1]. This algorithm is inspired by the growth and proliferation of the weeds. It operates based on the theory of choice and survival applying two strategies of k and R. Artificial grasses, or solutions, are selected at the start of the algorithm by implementation of R strategy of choice, where gradually the selection strategy changes to K. In order to introduce these two selective approaches in the nature, a brief explanation of these two strategies is provided:

R Selection Strategy: “immediate proliferation, rapid growth and death in youth”

The R selection strategy is the strategy of success and survival in unstable and unpredictable environments, in which reproduction is opportunistic and quick. The attributes to consider in R selection are high fertility, small size, and the ability to disperse over long distances.

K Selection Strategy: “slow life, slow reproduction, and death in aging”

The K selection strategy is the strategy of success and survival in the stable and predictable environments, where there is an intense competition among the individuals to attain limited resources. These conditions exist in the populations whose size is close to the habitat saturation. The attributes to consider in the K selection strategy are large size, long life, and small numbers of intensively cared-for offspring.

3.1. Invasive Weed Optimization (IWO)

The steps of implementing the invasive weed evolutionary algorithm are as follow:

1- A limited number of grains are scattered throughout the search space.

2- Each grain grows and produces a number of seeds according to its fitness.

3- The produced seeds are randomly dispersed in the problem space, and grow new plants.

4- These steps continue until the number of plants reaches the maximum value. Now the plants with a higher fitness are left, and the rest are removed.

These steps are repeated until the maximum iteration, and finally, the plant with the best fitness is selected. Each member of the plant population produces seeds based on the lowest and highest amount of colony as well as its related fitness and maximum number of seeds allowed. In other words, as the amount of fitness increases, the number of allowed seed production grows linearly to the maximum number of seeds (Figure 1). Thus undesirable plants can also reproduce, and if they contain useful information, it will not be lost.

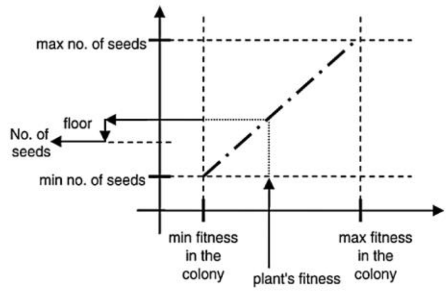

Figure 1. Number of produced seeds.

3.2. Spatial Dispersal

In this section, the issue of randomness and adaptation is considered in the algorithm. Using the normal distribution with the mean zero and variable variance, the produced seeds are randomly scattered in the d-dimensional space. This means that the seeds are randomly scattered near their parents. On the other hand, σ of the function declines from a pre-defined initial value to a final value in each iteration.

$$\sigma_{iter} = \frac{(iter_{\max}\text{-}iter)^n}{(iter_{\max})^n}(\sigma_{initial} - \sigma_{final}) + \sigma_{final} \qquad (1)$$

Thus the propagation space of seeds decreases non-linearly in each iteration. As a result, R selection changes into K selection.

3.3. Competitive Exclusion

After several iterations, the number of plants reaches its maximum but it is expected that the plants with a greater merit produce more than undesirable ones. When the population reaches the maximum number of plants, the process of removing plants with low competence begins. Now every plant is allowed to produce seeds, and the seeds are released into space. The seeds are then ranked along with their parents by fitness, and the plants with less fitness are removed until the colony reaches its permissible population.

4. Image Segmentation using Local Thresholding

The image segmentation methods can be divided into 6 different categories: edge-based methods, histogram-based methods, physical model-based methods, region-based methods, methods based on neural networks, and methods based on clustering. The histogram-based segmentation methods are divided into three categories: the methods that use a global threshold, the methods that use more than one threshold, and those that use local thresholds. The first category is suitable for the images with only two classes of intensity. The threshold value in these images is chosen as the lowest point of the histogram curve [2-6]. The second category splits the dynamic range of image into smaller ranges by selecting two or more thresholds. Then each range of intensity is assigned a specific color, which is called multi-level thresholding. In most of these methods, the valleys of the histogram are automatically found, and the lowest point of each valley is chosen as a threshold [7]. The above two methods use the global histogram and search for the global and local minima to assign pixels to their own classes. However, in images with lots of detail, these methods do not work well. Further, small objects are not well-segmented in these methods. The third category uses the local histograms for image segmentation and binarization. In these methods, the image is scrolled through windows of a certain size. At each step, the mean and standard deviation of intensity values of window pixels are obtained [8]. Then a single threshold is calculated for that window, and that part of image becomes binary. Thus in this method, the image boundaries are calculated more precisely. In these methods, the local thresholds are commonly derived from the following equation:

$$T_{xy} = a\sigma_{xy} + bm_{xy} \tag{2}$$

where σ is the standard deviation, m is the local mean, and *a and b* are two constants.

In the binarization with local thresholds, if the window size is smaller than the object, unevenness in the objects also becomes apparent, as the window threshold is calculated regardless of the edges. On the other hand, if the window size is larger than the object, it is possible to remove that object by thresholding. It is because the threshold is calculated according to the total values in the window. Meanwhile, the correct selection of the values of *a* and b greatly reduces the undesirable edges. Also smoothing the image can reduce the edges inside the objects. In our proposed method, the optimal parameters of local thresholding and smoothing are selected using the invasive weed algorithm.

5. Proposed Method

The proposed algorithm for object segmentation consists of the two main stages of edge detection and texture analysis.

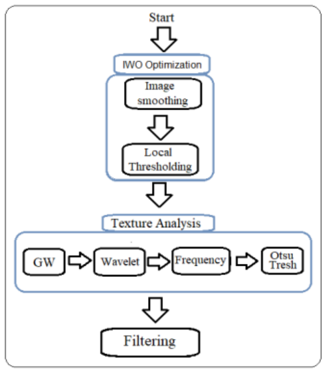

Figure 2. Proposed method steps.

5.1. Edge Detection

In this section, the invasive weed evolutionary algorithm and the local thresholding method are used for edge detection. For a better edge detection by local thresholding and for avoiding the discovery of extra edges, first the Gaussian smoothing is carried out. The smoothing filter multiplies the following equation by image intensity at the x location of the Gaussian sliding window:

$$\mathcal{N}(x \mid \mu, \sigma^2) = (1 / (2\sigma^2)^{1/2}) exp\left\{-(1/2\sigma^2)(x\_\mu)^2\right\} \tag{3}$$

where μ represents the mean of the filter and is equal to zero, and σ denotes the standard deviations of the filter; x also depends on the size of the window. Thus the smoothness of the filter depends on the arguments σ and the size of the window. The amount of smoothing for each image should be adequate for it. For example, the high-contrast images require more smoothness. Generally, the smoothness of an image depends on the amount of noise, contrast, and resolution. The parameters we require in binarization are the coefficients *a and b* mentioned in Equation (2), setting the desired thresholds. The coefficients have to be specified for each image separately. The values of these two parameters also depend on the contrast, degree of brightness of the image, and intensity changes in the image. Thus by the edge detection of each image, the four parameters *a, b,* σ*,* and size of the Gaussian window *(h)* must be specified for that image. It is clear that doing this manually is very difficult. As such and to calculate them, we used the invasive weed algorithm.

5.1.1. Edge Detection using Invasive Weed Algorithm

The edge detection operation involves the two steps of extracting the fine and thick edges, where each one of these two steps requires the four parameters mentioned. Hence, the number of unknown parameters for edge detection is eight parameters. The overall fitness value is obtained from the fitness factors. For each one of the fine and coarse edge detections, the fitness factors are calculated separately and the overall fitness value is obtained using two sets of factors. The thin edges are the edges that have a poor contrast to the background so the variance of the image at those points is small. This contrast reduction can be due to several reasons such as placing an object in shadow or an intense light and having the same color as the background.

In order to implement the weed algorithm, the initial and maximum numbers of seeds were 10 and 25, respectively. Thus at the beginning of each iteration, 10 plants from the bests are selected, and at the end, 25 new seeds of them are produced. Therefore, the population tends into the better responses. Since we have 10 seeds at the beginning, there is enough diversity of new solutions. In order to avoid the computational complexity and have an appropriate variety, we selected the maximum number of 25 seeds. The initial population was randomly selected, and the numerical ranges chosen for each one of the fine and coarse edge detections were different. For pallid edges, 'a' was selected in the range of 3-15, 'b' from 0 to 1, 'σ' from 1 to 5, and h from 3 to 5. Also for thick edges, 'a' and 'b' were selected the same as above, 'σ' was selected in the range of 4-10, and h from 3 to 7.

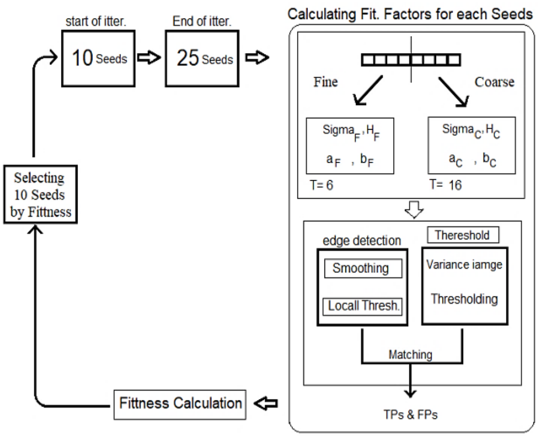

Figure 3. Operation cycle in each iteration.

The seeds of the plants grow around themselves, and they have similar properties to each other. In each iteration, the radius of the grain distribution area shrinks, whereby the seeds become more similar to their parents. The amounts of vectors produced in each iteration were checked, and if there were numbers outside the ranges, the range bounds would replace the random numbers.

5.2. Fitness Function

The fitness value is calculated using the original image variance. The procedure of fitness calculation is as follows:

First, an edge image is multiplied by the variance image. In that case, the white pixels will have the same amount as their corresponding pixel value in the variance image. It is clear that the variance image has a larger amount in the edge regions, and this is definitely larger in the thick edges' location. Accordingly, we can measure the accuracy of an edge detection. In order to calculate the fitness, we set two thresholds for the variance values appearing in the two edge images. We set a threshold of 6 in the delicate edge detection, while for the coarse edge detection, we set a threshold of 16 for them. This means that the points below these thresholds are considered *FPs* (False Positives), and the others *TPs* (True Positives). Note that in the delicate edge detection, the thick edges are not considered *FP*. In order to obtain the first set of fitness factors, from the non-zero pixel values, we considered the values greater than the threshold of 6 as *TPs*, and the others as *FPs*. Also in order to determine the second set of fitness factors, we find the values greater than threshold 16 as *TPs*, and the values below it as *FPs*. The thin edges are *Fps* here. The first fitness function we used to evaluate the edges was:

$$Fit = \frac{fine\ edges\, TPs + coarse\ edges\ FPs}{fine\ edges\ FPs + coarse\ edges\ FPs} \qquad (4)$$

It means that the fitness value is equal to the total number of *TPs* in both the fine and coarse edge detections divided by the total number of *FPs* in them.Note that due to the undesirable smoothing in each one of the binarized images, it is possible that very few edges are obtained. In this case, the pixels obtained are correct but the edge detection is incomplete. (Remember that the edge detection performance is not only dependent on the level of smoothing but also depends on the thresholding parameters.) Due to the low number of *FPs*, the above equation gets a good fitness for them.

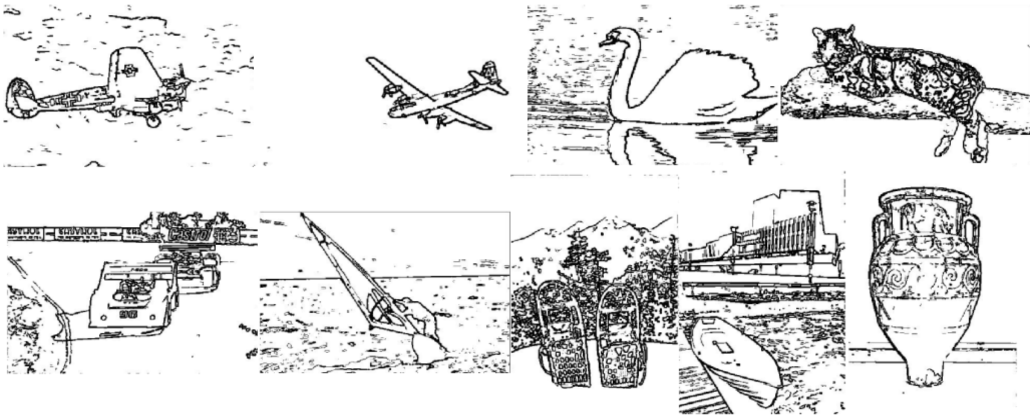

Figure 4. Edge detection results

In order to prevent this, we add another parameter to the fitness calculation. We used the size of maximum side of image to evaluate the edges. As a result, the algorithm's fitness function is as follows:

$$Fit = \frac{fine\ edges TPs + coarse\ edges\ FPs}{fine\ edges\ FPs + coarse\ edges\ FPs \times max\ side} \quad (5)$$

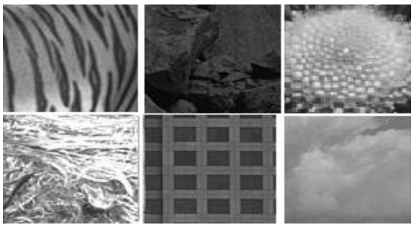

Figure 5. Internal texture of objects.

5.3. Texture Analysis

At this stage, more texture edges within the objects are removed by texture analysis. The texture edges are not the thin edges that have been found in the previous step because those thin edges can be part of the object's outer edges and a major edge in object segmentation. However, they have a low variance due to their poor contrast with the background. While the edges of the internal texture of the objects is located in the high frequency areas, they are very close to each other, and can even have a high variance.

.

5.3.1. Texture Analysis using Gradient Weight

The Gradient Weight (GW) transform gives each pixel a weight based on the inverse value of the image gradient in its place. In this picture, the pixels with a small gradient magnitude have a large value, and the pixels with a large gradient magnitude have a small value. As the texture edges have similar properties in each particular texture, they will have the same weight. Thus each different texture of the image is given a specific gray value in this transformation (Figure 7).

Note that in the GW image, the outer edges of the objects have a very low value due to their high contrast with the background. According to these facts, this transform makes an excellent separation of the different textures. Afterward, we used the Gaussian filter, and then opening for removing part of the inner edges.

5.3.2. Texture Analysis using Wavelet

Another function we used in order to extract the outer edges was the db1 wavelet. We applied the plural, vertical, and horizontal wavelet, respectively, on the output of the previous stage. In the case of plural wavelet, we obtained a good approximation of the outer boundaries applying the '*Range'* filter. This filter sets the central pixel value of a neighborhood for subtraction of minimum from the maximum value in that neighborhood. Therefore, the edge regions where the intensity difference increases will have a higher pixel value. In the case of horizontal and vertical wavelets, we removed the edges smaller than 20 pixels, and then ran opening by line structural elements to get the enclosed sections locating between the horizontal and vertical main lines.

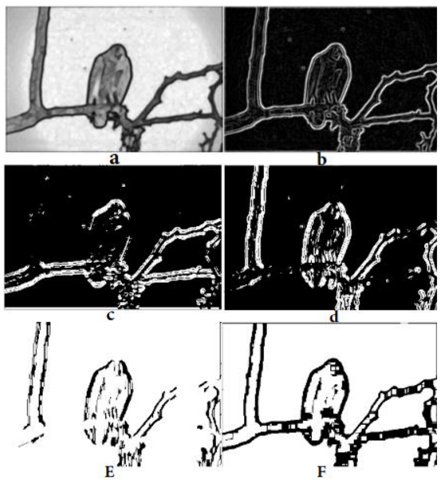

Figure 6. (a) Plural wavelet. (b) Output of range filter on (a). (c) Output of range filter on horizontal and (d) vertical wavelet. (E) Output of opening on (d). (F) Multiplying result of opening on (c) and (d).

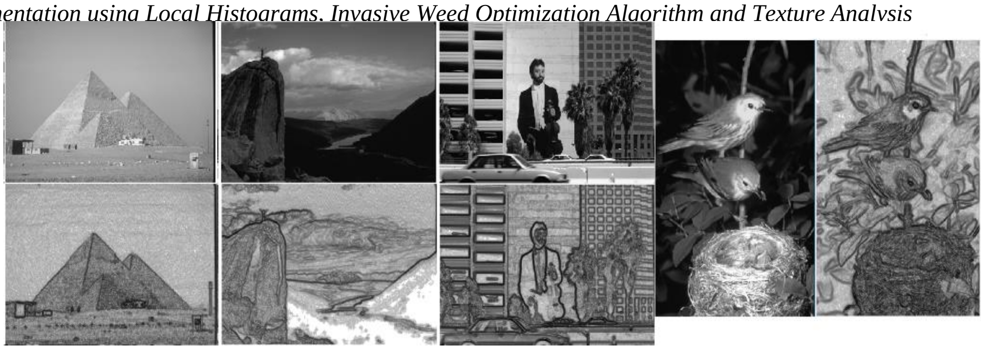

Figure 7. Gradient weight transform.

An approximation of the connected outer edges was obtained, multiplying the two wavelet images (Figure 6). Through multiplying these images by the invasive weed algorithm output image, more favorable edges were obtained.

5.3.3. Image Frequency Analysis

The low-frequency projection of an image consists of the smooth, non-periodic, and invariable parts of it. In this image, the textured edges of the objects including the fine and high-frequency edges will be eliminated. The level of elimination depends on the cut-off frequency.

Thus we set the rate below 20 Hz for it. In the output image, the edges are caused by the intensity variation between the objects and their background, and their extraction is effective to find the outer edges (Figure_8(d)). Then as with the previous step, we used the local '*range'* filtering for the binarization and bolding edges. The use of the gradient of this output image can be just as useful as a high-frequency filter. Obtaining the common edges in the outputs of the wavelet and frequency analysis can remove the redundant edges. As a result, a binary mask was created (Figure 8(E)).

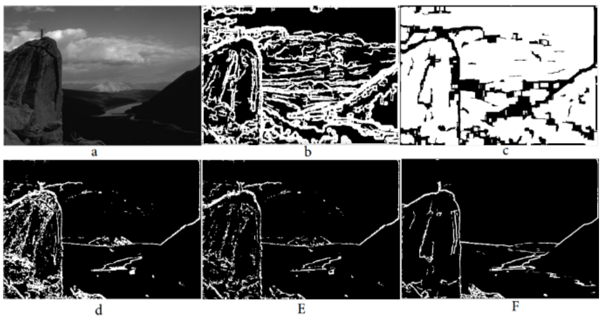

Figure 8. (a) Original image. (b) Binary output of plural wavelet. (c) Edges resulting from vertical and horizontal wavelets. (d) Binary output of frequency analysis. (E) Binary mask. (F) E multiplied by output image of edge detection phase.

5.3.4. Utilization of Otsu threshold

As mentioned earlier, the GW image has a value close to zero at the main edges. Using this property and the mask produced in the previous step, we find the main edges.

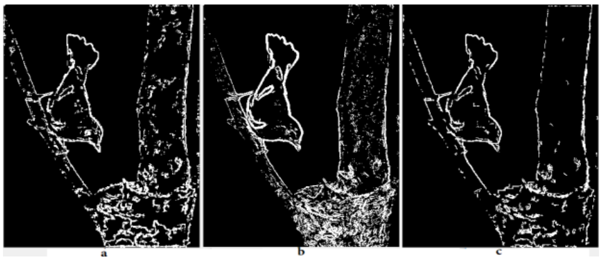

Figure 9. (a) The binary mask. (b) result of Otsu method. (c) a multiplied by b.

That way, we obtain the points matching the mask in the GW image, and the histogram of them as well as the Otsu threshold.

The matching points have a smaller intensity, and then the Otsu's threshold separates the darkest parts of them. These points can be approximated to the outer edges. The Otsu output has some fractures, and to improve it, we use the distance transform and watershed. We then multiplied the binary mask by the watershed output.

5.3.5. Gradient and Low-frequency Filtering

Low-frequency filtering of the gradient image can also be useful. We separated the values above 0.1 in the gradient image (we removed the smooth and texture-free areas), and then obtained the frequencies below 20 Hz. The low-frequency areas include the fine-texture areas such as the tree leaves, fine striped, and checkered textures. The gradient filtering connects them together, and thresholding produces uniform white areas (Figure 10). That way, we dim the white areas by a fraction of 0.3, and add to the entropy image and then threshold it. Finally, we obtain the common edges of the invasive weed edge detection and this image.

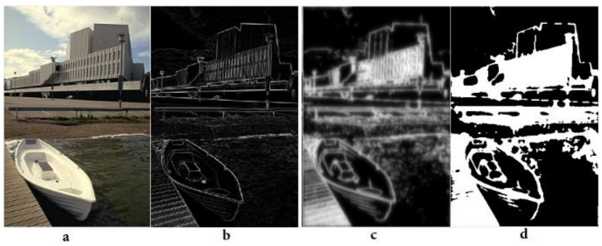

Figure 10. (a) Original. (b) Gradient image. (c) Low-frequency of (b). (d) Binary white areas.

5.4. Output Images of Object Detection Algorithm

In the following page, you can see the segmentation results of our proposed algorithm. Also we compare them with some results of the other methods.

Our algorithm can segment most of the natural images of the dataset and objects that have a normal situation in them. However, in those images that include non-uniform and severe lighting, weak contrast, the object with multi-fold textures or several objects with the same texture and very small objects; it shows a poor segmentation. In general, our algorithm made a perfect segmentation on the images whose objects had a uniform inner texture and had a normal contrast on their background.

6. Evaluation and Comparison

In our previous method of edge detection of natural images [30], we used a genetic algorithm in the edge detection step. The values of *a*, b, **σ**, and h were obtained by evaluating and ranking each chromosome that include 4 mentioned parameters using the following equation:

$$Fit = number\ of\ ones\ in\ \left(edge\ image*Laplacian\ image\right) \tag{5}$$

where 'edge image' is the output of genetic algorithm, and 'Laplacian image' is the binary image resulting from Laplace transform on smoothed input image. Afterward, we run the texture analysis for Object Segmentation on edge detection outputs. We evaluated these two methods (genetics and weeds) with the Berkeley, BSDS500 dataset [31]. The dataset consists of 500 natural images and their ground-truths that have been segmented by the human observers. The images are in the size of 321 × 481 or 481 × 321, and are the jpg images. The comparison results are shown in the table below.

Table 1. Results of comparison of GA and IWO.

Algorithm	OIS_F _score	ODS_F _score	Precision	Recall
GA method	0.69	0.66	61%	79%
IWO method	0.72	0.71	67%	79%

ODS_F_Score: The value of F is obtained over the whole dataset, where F has resulted from the average Precision, Recall.

OIS_F_Score: The value of F derived from averaging the distinct images Fs.

AP: The Average Precision.

We also compared our method with other methods of object segmentation evaluated on the BSDS500 dataset [31] in the following table.

Table 2. Comparing our method with other methods.

Algorithm	BSDS500 dataset		
	ODS	OIS	AP
Human	0.80	0.80	----
Mean_shift [26]	0.64	0.68	0.56
NCuts [27]	0.64	0.68	0.45
Canny_owt_ucm [25]	0.60	0.64	0.58
Felz_Hult [28]	0.61	0.64	0.56
Canny	0.60	0.63	0.58
AFWA [33]	0.49	0.49	---
MS-PFCD [34]	0.70	0.71	0.71
Our method	**0.72**	**0.71**	**0.67**

By comparing the evaluation results, we can understand that the proposed algorithm has a high performance for object segmentation in natural images.

7. Conclusions

In this article, we presented a method for segmenting the objects in natural images. The proposed method segments the image objects using the invasive weed optimization (IWO) algorithm, local thresholding, and texture analysis. It works without any requirement for the training data. IWO is an evolutionary algorithm that operates based on the theory of choice and survival, applying the two strategies of k and R. The local thresholding method uses the local histograms for image binarization. In order to improve segmentation, the Gaussian smoothing was executed before thresholding. The parameters of the Gaussian smoothing and local thresholding were specified using the invasive weed algorithm. Finally, the edges of the inside texture of objects were removed using the texture analysis. The evaluation results declare that the proposed algorithm has a high performance for segmenting the objects in natural images.

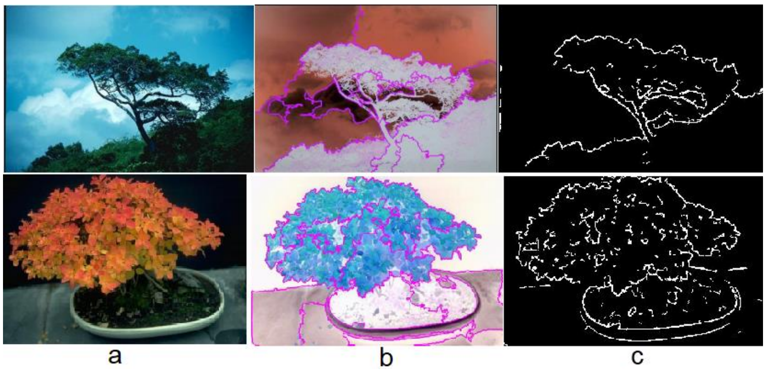

Figure 11.a) Original b) Segmentation by method [34] c) Segmentation by our method.

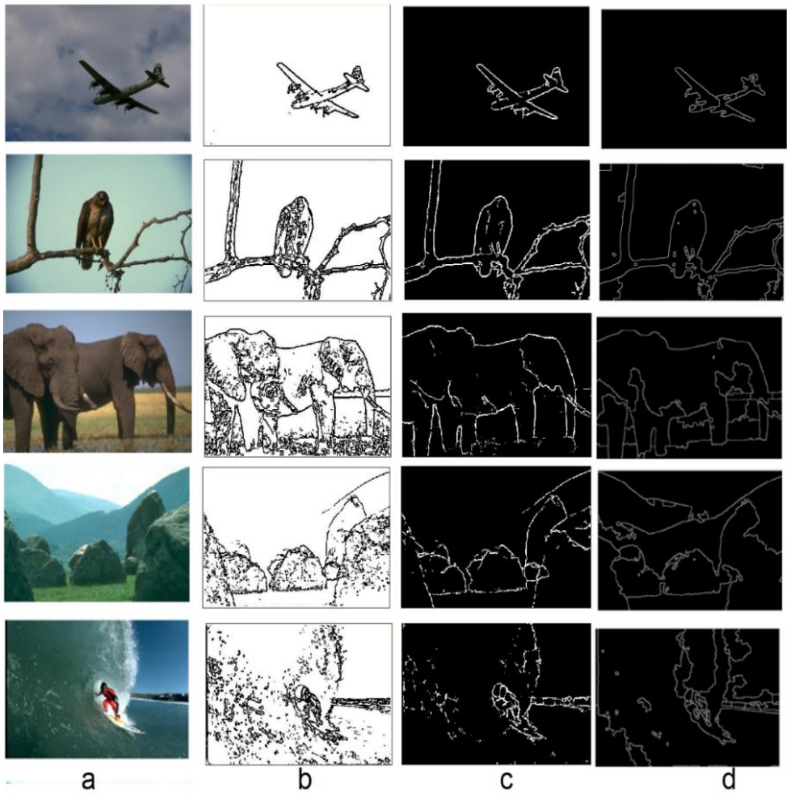

Figure 12. a) Original b) Edge detection C) Obj segmentation by our method d) Obj segmentation by [32].

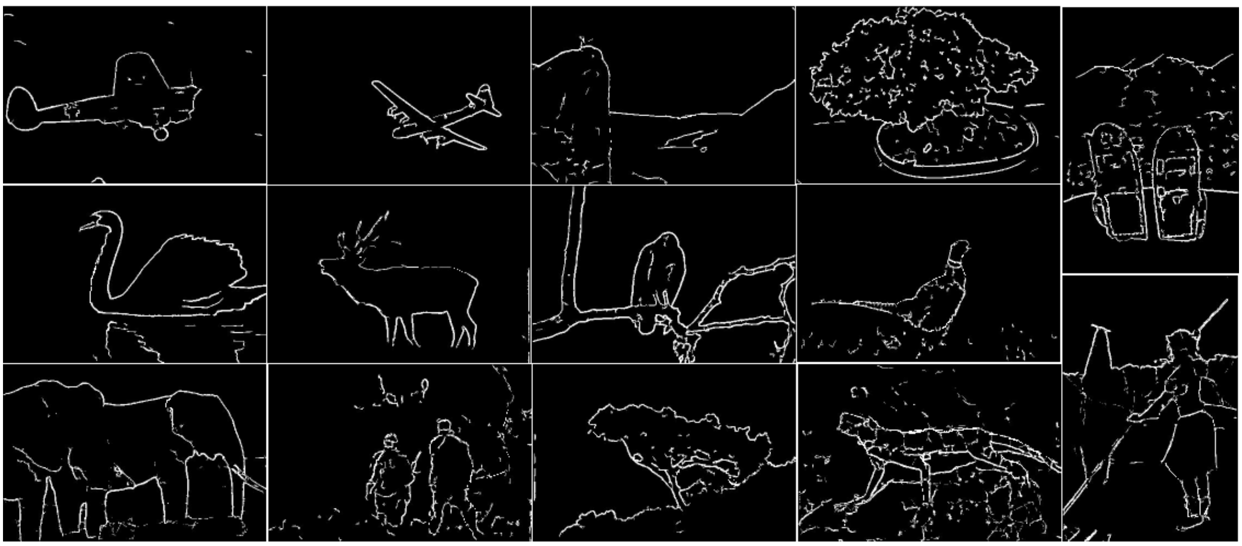

Figure 13. Output images of our object detection algorithm.

References

[1] A.R.Mehrabian, and C. Lucas, "A novel numerical optimization algorithm inspired from weed colonization" *Ecological Informatics1*, pp. 355-366, 2006.

[2] N. Otsu, "A threshold selection method from gray-level histogram" *IEEE Transactions on Man & Cybernetics*, Vol. 9, No. 1, pp. 62-66, 1979.

[3] G. Ribons, "Color edge detection", *Optical Engineering*, Vol. 16, No. 5, 1966.

[4] J.M.S. Prewitt and M.L. Mendelsohn, "The analysis of cell images" *Annals of the New York Academy Science*, Vol. 128, pp. 1035-1053, 1966.

[5] M.S. Al-tarawneh, "Lung cancer detection using image processing techniques" *Leonardo Electronic Journal of Practices and Technologies*, Issue 20, pp. 147-158, 2012.

[6] R. Rodriguez, "A Robust Algorithm for Binarization of Objects" *Latin American Applied Research*, Vol. 40, 2010.

[7] H. Jie, Z. Xiaojun, Y. Chunhua, and G. Weihua, "A multi-threshold image segmentation approach using state transition algorithm," in *Proceedings of the 34th Chinese Control Conference*, July, pp. 28-30, 2015.

[8] R.C. Gonzalez and R.E. Woods, "Intensity transformation and spatial filtering" in *Digital Image Processing*, 3rd ed. Prentice Hall, 2008.

[9] S. Rahnamayan, H.R. Tizhoosh, and M.M.A. Salama, "Robust Object Segmentation using Genetic Optimization of Morphological Processing Chains," in *Proceedings of the 5th WSEAS International Conference on signal, speech and image processing*, pp. 248-253, August 2005.

[10] K.S.N. Ripon, L.E. Ali, S. Newaz, and J. Ma, "A Multi-objective Evolutionary Algorithm for Color Image Segmentation" *Mining Intelligence and Knowledge Exploration. MIKE. Lecture Notes in Computer Science*, Vol. 10682, 2017.

[11] U. Kirchmaier, S. Hawe and K. Diepold, "A Swarm Intelligence Inspired Algorithm for Contour Detection in Images" *Applied Soft Computing*, Vol. 13, Issue 6, pp. 3118-3129, 2013.

[12] W. Fu, M. Zhang and M. Johnston, "Bayesian Genetic Programming for Edge Detection" *Soft Computing*, Vol. 23, Issue 12, pp. 4097-4112, 2019.

[13] W. Tao, H. Jin, and L. Liu, "Object segmentation using ant colony optimization algorithm and fuzzy entropy" *Pattern Recognition Letters*, Vol. 28, pp. 788–796, 2007.

[14] L. Xu, H. Jia, C. Lang, X. Peng, and K. Sun, "A Novel Method for Multi-level Color Image Segmentation based on Dragonfly Algorithm and Differential Evolution", *IEEE Access*, Vol. 7, pp. 19502-19538, 2019.

[15] X. Zhao, M. Turkb, W. Lid, K. Lien, and G. Wang, "A multi-level image thresholding segmentation algorithm based on two-dimensional K–L divergence and modified particle swarm optimization" *Applied Soft Computing*, Vol. 48, pp. 151-159, 2016.

[16] O. Banimelhem, and Y.A. Yahya, "Multi-thresholding Image Segmentation using Genetic Algorithm", In *World Congress in Computer Science, Computer Engineering, and Applied Computing*, 2011.

[17] Y. Zhang and L. Wu, "Optimal Multi-level Thresholding based on Maximum Tsallis Entropy via an Artificial Bee Colony Approach" *Entropy*, Vol. 13, pp. 841-859, 2011.

[18] C.C. Lai, "A Novel Image Segmentation Approach based on Particle Swarm Optimization" *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E89-A, No.1, pp. 324-327, 2006.

[19] P. Moallem, and N. Razmjooy, "Optimal Threshold Computing in Automatic Image Thresholding using Adaptive Particle Swarm Optimization" *Journal of Applied Research and Technology*, Vol. 10, No. 5, 2012.

[20] P.Y. Yin and L.H. Chen, "A fast iterative scheme for multi-level thresholding methods" *Signal Processing*, Vol. 60, pp. 305-313, 1997.

[21] S.A. Mohammadi, R. Akbari and S.H. Mohammadi, "An Efficient Method based on ABC for Optimal Multi-level Thresholding" *IJST Trans. Of Electrical Engineering*, Vol. 36, No. E1, pp. 37-49, 2012.

[22] J. Musavirad, and H. Ebrahimpour, "Optimal Multilevel Image Thresholding using an Optimization Algorithm based on Learning and Teaching" *Journal of Image Processing and Machine Vision*, 2nd Year, No. 2, 2015, Farsi.

[23] P. Kanungo, P.K. Nanda and , U.C. Samal "Image segmentation using thresholding and genetic algorithm," in *Proceedings of the Conference on Soft Computing Technique for Engineering Applications, SCT*, Rourkela, India, pp. 24-26, March 2006.

[24] J. Yang, Y. Yang, W. Yu and J. Feng, "Multi-threshold image segmentation based on K-means and Firefly Algorithm," in *3rd International Conference on Multimedia Technology, ICMT*, 2013.

[25] P. Arbelaez, M. Maire, C. Fowlkes and J. Malik, "Contour detection and hierarchical image segmentation" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 33 (5), pp. 898–916, 2011.

[26] D. Comaniciu, and P. Meer, "Mean shift: a robust approach toward feature space analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24 (5), pp. 603 –619, 2002.

[27] T. Cour, F. Benezit, and J. Shi, "Spectral segmentation with multi-scale graph decomposition," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1124–1131, 2005.

[28] P.F. Felzenszwalb and D.P. Huttenlocher, "Efficient graph-based image segmentation" *International Journal of Computer Vision*, Vol. 59 (2), pp. 167–181, 2004.

[29] M. Maire, P. Arbelaez, C. Fowlkes and J. Malik, "Using contours to detect and localize junctions in natural images" *CVPR computer Vision and Pattern Recognition*, 2008.

[30] M.H. Hasheminezhad and S. Bayatpour, "Image Segmentation using Local Thresholds and Genetic Algorithm," in *2nd International conference on computer engineering and information technology*, Tehran, Iran, 2017 [Farsi].

[31] P. Arbelaez, M. Maire, C. Fowlkes and J. Malik, "Contour Detection and Hierarchical Image Segmentation" *IEEE TPAMI*, Vol. 33, No. 5, pp. 898_916, 2011.
https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html.
doi = {10.1109/TPAMI.2010.161}

[32] R. Aslanzadeh, K. Qazanfari and M. Rahmati, "An Efficient Evolutionary based Method for Image Segmentation," arXiv:1709.04393, 2017 [online].

[33] M.H. Arsay, Suyanto, and K.N. Ramadhani, "Aerial Image segmentation with clustering using firework algorithm" *Indonesian journal of computing*, 2019.

[34] K.S. Ripon, L.E. Ali, S. Newaz, and J. Ma, "A Multi-objective Evolutionary Algorithm for Color Image Segmentation" *Lecture notes in computer science*, 2017.

[35] N. Widynski and M. Mignotte "A Multi-scale Particle Filter Framework for Contour Detection" *pattern analysis and machine intelligence*, Vol. 36, No. 10, 2014.

[36] Z. Dorrani and M.s. Mahmoodi "Noisy images edge detection: Ant Colony Optimization algorithm" *International Journal of AI and Data Mining*, Vol.4, pp. 77-83, 2016.

بخشبندی اشیاء با استفاده از هیستوگرامهای محلی، بهینه سازی علفهای هرز و تحلیل بافت

سمیه بیات پور و سید محمدحسین هاشمی نژاد*

گروه هوش مصنوعی، دانشگاه الزهراء، تهران، ایران.

ارسال ۲۰۲۰/۱۰/۲۷؛ بازنگری ۲۰۲۱/۰۵/۰۷؛ پذیرش ۲۰۲۱/۰۶/۲۹

چکیده:

بیشتر روشهایی که برای بخشبندی اشیاء در تصاویر، به‌کار می‌روند؛ روشهایی نظارتی هستند که به دلیل نیاز به داده‌های برچسبدار بسیار زیاد، روشهایی هزینه‌بر هستند. اما در این مقاله، ما روشی را برای بخشبندی اشیاء ارائه داده‌ایم که مبتنی بر بهینه‌سازی متا-هیوریستیک می‌باشد که نیازی به داده‌های آموزشی ندارد. این روش شامل دو مرحله اصلی: "لبه یابی" و "تحلیل بافت" است. در مرحله لبه‌یابی ما از بهینه‌سازی علفهای هرز و آستانه‌دهی محلی استفاده نمودیم. روشهای لبه‌یابی مبتنی بر هیسنوگرامهای محلی، روشهای بسیار موثر و کارایی هستند. اما تعیین پارامترهای مناسب برای آنها به‌طور دستی، بسیار مشکل است. بعلاوه، این پارامترها، باید برای هر تصویر، به‌طور خاص انتخاب شوند. در این مقاله روشی ارائه شده‌است که پارامترهای لازم برای لبه‌یابی را با بکارگیری یک الگوریتم تکاملی، به‌طور اتوماتیک تعیین می‌کند. نتایج ارزیابی این روش، بیانگر کارایی بالای آن بر روی تصاویر طبیعی می‌باشد.

کلمات کلیدی: بخشبندی اشیاء، آستانه محلی، هیستوگرام، بهینه سازی علفهای هرز، تحلیل بافت.