Expert Systems With Applications 65 (2016) 221–232

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A hybrid differential evolution for optimal multilevel image thresholding

Uroš Mlakar*, Božidar Potočnik, Janez Brest

Faculty of Electrical Engineering and Computer Science, University of Maribor, SI-2000, Slovenia

ARTICLE INFO

Article history:
Received 12 March 2016
Revised 11 August 2016
Accepted 12 August 2016
Available online 21 August 2016

Keywords:
Image segmentation
Differential evolution
Multilevel thresholding
Otsu criterion
Hybridization

ABSTRACT

Image thresholding is a process for separating interesting objects within an image from their background. An optimal threshold's selection can be regarded as a single objective optimization problem, where obtaining a solution can be computationally expensive and time-consuming, especially when the number of thresholds increases greatly. This paper proposes a novel hybrid differential evolution algorithm for selecting the optimal threshold values for a given gray-level input image, using the criterion defined by Otsu. The hybridization is done by adding a reset strategy, adopted from the Cuckoo Search, within the evolutionary loop of differential evolution. Additionally a study of different evolutionary or swarm-based intelligence algorithms for the purpose of thresholding, with a higher number of thresholds was performed, since many real-world applications require more than just a few thresholds for further processing. Experiments were performed on eleven real world images. The efficiency of the hybrid was compared to the cuckoo search and self-adaptive differential evolution, the original differential evolution, particle swarm optimization, and artificial bee colony where the results showed the superiority of the hybrid in terms of better segmentation results with the increased number of thresholds. Since the proposed method needs only two parameters adjusted, it is by far a better choice for real-life applications.

1. Introduction

Image segmentation is an operation within the field of image processing for separating interesting objects within an image from the background based on the gray-level distribution of the pixels. If the image is split into two classes it is called bi-level thresholding and is extended to multilevel thresholding for separating the image into more than two classes. Thresholding techniques can be categorized into two categories, namely global and local thresholding. In regard to global thresholding a single threshold value from the entire image histogram is selected, while local thresholding operates on smaller regions within the image and selects multiple thresholds, one for each region. When comparing the methods, global thresholding is simpler and also easier to implement but it relies heavily on good illumination of the input image. Local thresholding methods overcome this drawback but are also computationally expensive and harder to implement. Whether we choose the global or local category, the probability density of the gray-level histogram can be handled by either a parametric or a non-parametric approach in order to find the optimal threshold. For parametric approaches, being computationally expensive, the statistical parameters of the classes are estimated and are heavily dependent on the initial conditions. The non-parametric approaches choose thresholds based on maximizing certain criteria, like between-class variance (Otsu, 1975) or various entropy measures (Kapur, Sahoo, & Wong, 1985; Tsallis, 1988). In this work we adopted the between-class variance as the objective function for evaluating the efficiency of the selected thresholds.

In the field of image thresholding heuristic methods have gained a lot of attention, since the exhaustive methods are often inefficient or simply computationally expensive. Although obtaining good results, meta-heuristic methods like particle swarm optimization, ant colony optimization, etc. Fister, Yang, Fister, Brest, and Fister (2013), have their drawbacks, which can be resolved through hybridization (Mlakar, Fister, & Fister, 2016). These drawbacks are usually fast convergence to a sub-optimal solution (local optimum), poor segmentation results, when the number of thresholds increases greatly, and also the number of parameters which must be fine tuned prior to running the algorithm. The idea to overcome getting trapped in a local optimum is found in either adding domain knowledge of the problem to the search algorithm, or to hybridize the existing method with a mechanism to solve this problem. Hybrid methods exploit the good properties

* Corresponding author.
E-mail addresses: uros.mlakar@um.si (U. Mlakar), bozidar.potocnik@um.si (B. Potočnik), janez.brest@um.si (J. Brest).

http://dx.doi.org/10.1016/j.eswa.2016.08.046
0957-4174/

of different methods by joining them within an efficient solver for a given problem.

Significant work has been done on optimal threshold selection using evolutionary algorithms (Ayala, dos Santos, Mariani, & dos Santos Coelho, 2015; Maitra & Chatterjee, 2008), while many hybrids have also been proposed. Ali, Ahn, and Pant (2014) compared the efficiency of Synergetic Differential Evolution (SDE) using two different evaluation criteria, one being approximation on an image histogram by a mixture of Gaussian distributions and the other being Kapur's entropy (Kapur et al., 1985). Ayala et al. (2015) proposed a Beta Differential Evolution algorithm (BDE), to find the optimal thresholds of a given image by using Otsu's criterion. Horng and Liou (2011) compared swarm intelligence algorithms using minimum class entropy. They proposed a firefly-based hybrid, which performed better against particle swarm (PSO) and quantum PSO. Sarkar, Das, and Chaudhuri (2015) employed DE with the minimum cross entropy for testing on color images. Akay (2013) made a study on an Artificial Bee Colony (ABC) and PSO by comparing various criteria. It was concluded that both algorithms performed equally well when the number of thresholds was 2, but the ABC using the Otsu criterion performed better when the number of thresholds excelled. Ouadfel and Meshoul (2013) designed a fully adaptive hybrid PSO, where the thresholds' values are computed along with the number of thresholds to segment the image as well as possible. In a recent study by Ouadfel and Taleb-Ahmed (2016) they investigated two newer swarm intelligence algorithms, namely the Flower Pollination (FP) and Social Spider Optimization (SSO) algorithms to solve the image segmentation problem using Otsu's criterion and Kapur entropy. Their results indicate that both FP and SSO outperformed other algorithms used in the study. Hammouche, Diaf, and Siarry (2010) performed a study on six meta heuristic techniques, where the results pointed out that the DE gave better quality results, while the PSO converged more quickly. Dey, Saha, Bhattacharyya, and Maulik (2014) proposed six different quantum inspired meta-heuristic methods, which work as a consensus for the multilevel thresholding problem.

This article presents an efficient self-adaptive hybrid DE algorithm for the multilevel thresholding problem. The proposed method selects the optimal thresholds based on maximizing the between-class variance (Otsu, 1975).

The main contributions of this paper are two-fold: (1) The introduction of a hybrid self-adaptive DE algorithm (hjDE) for the optimal thresholds selection by maximizing the between-class variance, where the experiments show that hjDE achieved the best results, (2) To present a study of different evolutionary or swarm-based intelligence algorithms on the impact of thresholding with a higher number of thresholds. To our knowledge most of the research community focuses mainly on a smaller number of thresholds, while in many real-life applications of segmentation (i.e. deep-space photo analysis, object recognition, video compression, ...) more than just a few thresholds are needed. The proposed hybrid also needs just two parameters adjusted, which makes it favorable for use in real-life applications.

The rest of this paper is organized as follows. Section 2 presents the DE algorithm, and in Section 3 the cuckoo search is described. Image segmentation using Otsu's method for selecting the optimal thresholds is presented in Section 4. Section 5 is dedicated to the proposed algorithm, within an experimental environment, using the images, while efficiency metric and individual encoding are presented in Section 6. In Section 7 the numerical and statistical results are presented and discussed. The paper is concluded with future work directions in Section 8.

2. Differential evolution

Differential Evolution (DE) (Storn & Price, 1997) is a population based evolutionary algorithm used for continuous global optimization. It is simple, yet very effective at solving various real life problems. It has been modified and extended several times with various versions being proposed (Brest, Greiner, Boskovic, Mernik, & Zumer, 2006). The idea of DE is a simple mathematical model, which is based on vector differences.

The DE algorithm evolves a population of vectors, which represent the solutions to the problem, through generations. The population consists of Np vectors, where $i \in [1, \ldots, Np]$. In each generation depicted as g, a vector $\mathbf{x}_i^g = (x_{i,1}^g, x_{i,2}^g, \ldots, x_{i,D}^g)$, goes through a set of evolutionary operators, namely mutation, crossover and selection. A trial vector is produced using these operators, which competes with its parent based on fitness value for surviving into the next generation. Mutation is carried out for each population vector, which yields a mutant vector denoted as $\mathbf{v}_i^{g+1}$, using a selected mutation strategy. Many mutation strategies have been proposed but for the purpose of this paper the "rand/1" is used:

$$\mathbf{v}_i^g = x_{r1}^g + F\left(x_{r2}^g - x_{r3}^g\right). \tag{1}$$

Indexes $r1 - r3$, represent the random and mutually different integers generated within the interval $\{1, \ldots, Np\}$, while also being different from index i. F is a mutation scaling factor, which lies within the range [0, 1].

When the mutant vector $\mathbf{v}_i^{g+1}$ is generated, crossover is performed using the crossover rate Cr and the corresponding vector $\mathbf{x}_i$ from the population as follows:

$$u_{i,j}^g = \begin{cases} v_{i,j}^g, & \text{if } rand(0,1) \le C_r \text{ or } j = j_{rand}, \\ x_{i,j}^g, & \text{otherwise.} \end{cases} \tag{2}$$

The crossover rate Cr is defined within the interval [0, 1] and it defines the probability of creating the trial vector parameters $u_{i,j}^g$. The j_{rand} index is responsible for the trial vector containing at least one value from the mutant vector $\mathbf{v}_i^g$. After the crossover, some values of the trial vector may fall out of bounds, meaning they must be mapped to the defined search space. We employed the following strategy (Zhang & Sanderson, 2009):

$$u_{i,j}^g = \begin{cases} \dfrac{(lbound_j + x_{i,j}^g)}{2}, & \text{if } u_{i,j}^g < lbound_j, \\ \dfrac{(ubound_j + x_{i,j}^g)}{2}, & \text{if } u_{i,j}^g > ubound_j. \end{cases} \tag{3}$$

lbound and **ubound** are vectors containing the lower and upper bounds of the problem. After the trial vector is repaired, the selection process takes place:

$$\mathbf{x}_i^{g+1} = \begin{cases} \mathbf{u}_i^g, & \text{if } f(\mathbf{u}_i^g) \ge f(\mathbf{x}_i^g), \\ \mathbf{x}_i^g, & \text{otherwise.} \end{cases} \tag{4}$$

Each trial vector $\mathbf{u}_i^g$ competes with its parent $\mathbf{x}_i^g$ based on their fitness value f. The one with the better fitness value survives and is transferred to the next generation (Eq. (4)).

In a study by Neri and Tirronen (2010) it was concluded that a DE extension named jDE (self-adaptive differential evolution) performed better in terms of robustness and stability, where the main difference between the algorithms was the ways in which the control parameters behaved. In the original DE the scaling factor F and crossover rate Cr are constant, while in jDE (Brest et al., 2006) they are self-adaptive. In this paper we built a hybrid based on the jDE algorithm for threshold selection.

3. Cuckoo search

Cuckoo Search (CS) is a nature-inspired population based algorithm developed by Yang and Deb (2009). It belongs to the swarm intelligence based algorithms as it mimics the natural behavior of some cuckoo species and their brood parasitism. These cuckoo