

Multilevel threshold selection for image segmentation using soft computing techniques

C. Mala¹ · M. Sridevi¹

Published online: 1 May 2015
© Springer-Verlag Berlin Heidelberg 2015

Abstract Multilevel thresholding is the method applied to segment the given image into unique sub-regions when the gray value distribution of the pixels is not distinct. The segmentation results are affected by factors such as number of threshold and threshold values. Hence, this paper proposes different methods for determining optimal thresholds using optimization techniques namely GA, PSO and hybrid model. Parallel algorithms are also proposed and implemented for these methods to reduce the execution time. From the experimental results, it is inferred that proposed methods take less time for determining the optimal thresholds when compared with existing methods such as Otsu and Kapur methods.

Keywords Multilevel thresholding · Segmentation · Parallel environment · Genetic algorithm · Particle swarm optimization · Hybrid method

1 Introduction

Image segmentation is a process of grouping the pixels of the image based on some criteria such as intensity, color, texture, etc. Image segmentation is still a challenging problem in the field of computer vision and image processing. Edge detection, region-based segmentation and thresholding methods are most popularly used for image segmentation problems.

Edges or boundaries of the images (Sharif et al. 2002; Kaur et al. 2011) are computed based on the dissimilarity property of the image. Segmentation process using region (Peng et al. 2011) or thresholding method (Otsu 1978; Kapur et al. 1985; Gonzalez and Woods 2008) uses homogenous criteria to segment the image into distinct regions.

Thresholding is a simplest method, which partitions the image into meaningful homogenous sub-regions using threshold values. The number of threshold values used for image segmentation varies depending on the application or image. The best threshold number and values are obtained by trial and error method. The result of segmentation is given as an input for high-level processing such as recognition, matching, extraction, analysis, etc. The selected threshold value should be appropriate else it will affect the result of high-level processing. Hence, the threshold selection should be optimal. Thresholding methods are divided into two types based on the number of threshold values namely,

1. Bilevel thresholding—uses one threshold value. It forms two regions: one below the specified threshold and another region above the selected threshold.
2. Multilevel thresholding—uses more than two threshold values. It forms ' $n + 1$ ' homogenous sub-regions for ' n ' threshold values.

The thresholding methods use parametric or nonparametric approach. The parametric approach depends on gray distribution of the pixels; whereas, the nonparametric approach, optimizes the objective function to find out the threshold values. Both approaches are computationally expensive and time-consuming (Zahara et al. 2005). Hence, they are unsuitable for real-time applications and there is a need for fast and robust method to solve multilevel thresholding problems.

Communicated by A. Jara, M. R. Ogiela, I. You and F.-Y. Leu.

✉ C. Mala
mala@nitt.edu

M. Sridevi
msridevi@nitt.edu

¹ Department of Computer Science and Engineering, National Institute of Technology, Tiruchirappalli 620 015, India

Many methods were proposed in the literature to determine multiple thresholding values for the images which do not have distinct range of intensity distributions. Even though good results are produced by methods proposed by Yin and Chen (1997); Hammouche et al. (2008); Zhang et al. (2011); Sridevi et al. (2014), etc., they require huge computational time due to their iterative process.

2 Related works

Image segmentation is the fundamental step in image analysis, understanding, retrieval, recognition, interpretation, etc. It is a process of partitioning the image into multiple segments based on some criteria such as intensity, color, texture, etc. for proper recognition of objects in the image. Thresholding is a method used to segment the image. An optimal value of threshold is to be chosen to separate one or more desired objects from their background and is application dependent. Determination of optimal threshold values remains as a challenging problem in image segmentation.

Iterative scheme is proposed by Yin and Chen (1997) to determine an appropriate threshold value. The scheme is iterated until the threshold values are not changed. The disadvantage of the proposed method is that it strikes on local optimal value when more than three thresholds are obtained.

To address the thresholding problem, different criteria such as Automatic Thresholding Criterion (ATC) (Cheng et al. 1995), maximum correlation criterion (Djerou et al. 2012), maximum entropy criterion, maximum fuzzy entropy (Lan and Zeng 2013), class variance (Otsu 1978) are proposed in the literatures. More criteria are to be considered to have better segmentation results for all kind of images.

Otsu method (1978) uses between-class variance of the image histogram as an objective function. The method maximizes the objective function to get optimal threshold values. The method does not work well for darker background and noisy images. The standard Otsu is combined with either Fuzzy entropy (Lan and Zeng 2013) or morphology operations (Zhang et al. 2011) to get better results.

ATC is used by Cheng et al. (1995) for solving multilevel thresholding problem by minimizing the function. The automatic threshold criterion uses less computational complexity than the entropy criterion (Lan and Zeng 2013).

The computational complexity of methods such as Otsu, Kapur, entropy criteria, ATC, etc. is more due to exhaust search performed by those algorithms. To reduce the complexity, optimization techniques such as Genetic Algorithm (GA) (Hammouche et al. 2008; Indira and Ramesh 2011; Kumar et al. 2012; Sridevi et al. 2014), particle swarm optimization (Djerou et al. 2012; Osuna-Enciso et al. 2013), differential evolution algorithm (Pei et al. 2009; Ali et al. 2014), bee colony optimization (Akay 2013), intelligent

techniques (Sathya and Kayalvizhi 2012) have been widely applied. The objective of an optimization problem is to compute a value for the variables such that it should satisfy the given constraints and objective function. Evolutionary algorithms combined with wavelet transformation (Hammouche et al. 2008; Kumar et al. 2012; Sridevi et al. 2014) are used in the literature for image segmentation problem to run faster. ATC function can be minimized by applying GA. Hybrid models (Zahara et al. 2005; Ouadfel and Meshoul 2013) are also employed for solving the multilevel threshold selection.

The following are certain drawbacks of the existing methods:

1. It is difficult to find an optimal threshold.
2. Computation and space complexity of the algorithms are relatively high.

To address these issues, this paper proposes an Optimal Threshold Selection (OTS) using optimization techniques such as GA, PSO and hybrid methods. The objective function used is a minimization function and is coined based on criteria such as variance between the classes, within the class, goodness, thresholding criterion and segmentation criterion to suit all type of images. The proposed methods provide optimal threshold values and further the computational complexity of these proposed methods are reduced by executing the algorithms in a parallel environment. The input image is decomposed into 'j' levels using Discrete Wavelet Transformation (DWT) which in turn reduces the space complexity of the proposed algorithms.

The remainder of the paper is organized as follows: Sect. 3 explains the proposed Optimal Threshold Selection algorithms. The experimental results and their performance analysis are discussed in Sect. 4. Final Sect. 5 concludes the paper.

3 Proposed method

Recently image processing applications are widely used in many fields such as medical, defense, telemedicine, scientific publication, digital forensic, etc. Image segmentation is a process in which the given image is grouped into different regions based on homogenous or heterogeneous properties of the image. Segmentation is a preprocess or low-level process for applications such as object detection and recognition, image analysis and understanding, diagnosis of diseases, tracking of object, image classification, image retrieval, etc. Thresholding is a method of segmenting the image based on selected threshold values. This results into homogenous sub-regions. The threshold value is selected based on the peaks and valleys of the histogram. The histogram is a plot which shows the intensity distribution of the given gray image.

Threshold values may be single or multiple depending on the gray values of the image. The number of threshold and their values should be optimal. It is still a challenging task to find optimal thresholds for image segmentation problem. Some of the image criteria such as selection, segmentation, entropy, class variance, automatic threshold criterion, etc. are incorporated in the literature (Otsu 1978; Cheng et al. 1995; Djerou et al. 2012; Lan and Zeng 2013) to get better thresholds and segments. For real-time imaging applications, fast selection of optimal multilevel thresholds is required.

When the foreground and background pixel intensity distributions are distinct, it is sufficient to use single threshold for segmenting the object from the background. If this intensity distribution is not distinct, single threshold is not useful. It needs multilevel thresholds to segment the image. It is difficult to solve segmentation problems which require more than two threshold values. Existing conventional methods consume more time to determine the threshold values. To reduce the time, this paper proposes different algorithms by using various optimization techniques namely, GA, PSO and hybrid method and executes them in both sequential and parallel environments to solve multilevel thresholding problem on DWT of the image. The usage of optimization techniques such as GA and PSO for this work has the following advantages over traditional searching techniques.

- They are global searching techniques.
- They prevent trapping of local optimal solution.
- They determine optimal number of thresholds.

The following algorithms are proposed in this paper to segment the image using optimal selection of threshold values.

- OTS using Sequential GA (SGA)
- OTS using Sequential PSO (PPSO)
- OTS using Parallel GA (PGA)
- OTS using Parallel PSO (PPSO)
- OTS using Hybrid GA (HGA)
- OTS using Hybrid PSO (HPSO)
- OTS using Hybrid Parallel GA (HPGA)
- OTS using Hybrid Parallel PSO (HPPSO).

The proposed algorithms take gray image as an input. Hence, color images are to be converted into gray scale images. The gray image is decomposed into j levels using DWT to enable the algorithm to run faster. The image decomposition process is explained in Sect. 3.1. A new Combined Objective Function (COF) is coined and is used in GA, PSO and also for hybrid methods to analysis their performance. The derivation of COF is explained in Sect. 3.2. Sections 3.3 and 3.4 explain threshold selection using the sequential and parallel versions of GA and PSO, respectively. The subsequent Sect. 3.5 discusses about the hybrid model. They are used to

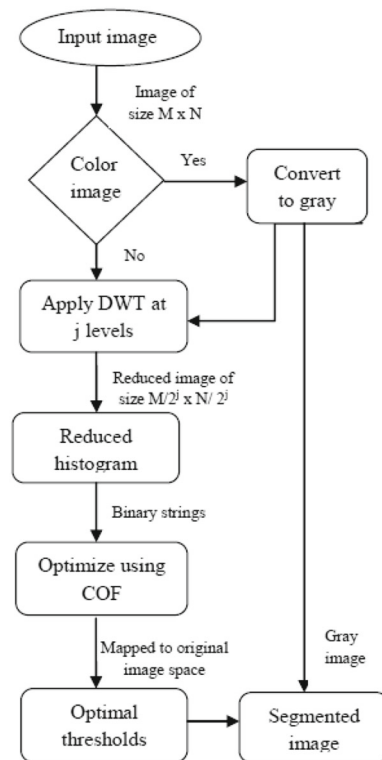


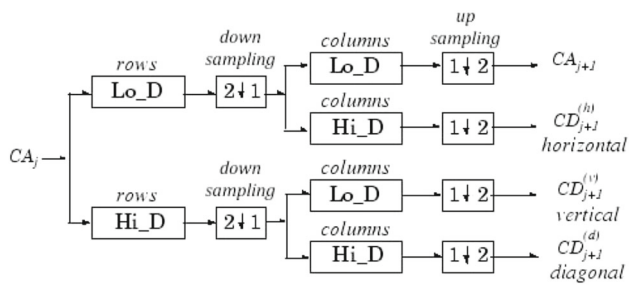
Fig. 1 Flow diagram depicting determination of an optimal threshold

select number of thresholds and their values for segmenting the given image. The hybrid model provides better results than the standard GA and PSO. The use of hybrid model for multilevel threshold selection gives proper threshold values which in turn segments the image into appropriate regions. The steps involved in the proposed Optimal Threshold Selection problem are shown in Fig. 1.

3.1 Image decomposition

To solve multilevel thresholding problem in a faster manner, the size of the original image is reduced. The computational complexity of the algorithms mainly depends on the size of the image. Hence, it has to be reduced by using DWT (Kim et al. 2003; Hammouche et al. 2008). The image is decomposed to ' j ' levels and it results in four components namely, approximation and detail coefficients in three orientations (horizontal, vertical, and diagonal). The general process of DWT decomposition (Gonzalez and Woods 2008; Strang and Nguyen 1996) is shown in Fig. 2. An example is shown in Fig. 3 for image decomposition along with different levels of decomposition components.

After applying DWT to the image of size $M \times N$ at j levels, the size of the image is reduced to $M/2^j \times N/2^j$. If more levels are used for decomposition of the image, then the computational complexity will be reduced. The thresh-



where,

Lo_D is Low pass filter Decomposition

Hi_D is High pass filter Decomposition

CA_j is Approximated coefficient at level j

CD_{j+1}^h is Horizontal decomposed coefficient at level $j+1$

CD_{j+1}^v is Vertical decomposed coefficient at level $j+1$

CD_{j+1}^d is diagonal decomposed coefficient at level $j+1$

Fig. 2 Decomposition using Discrete Wavelet Transformation

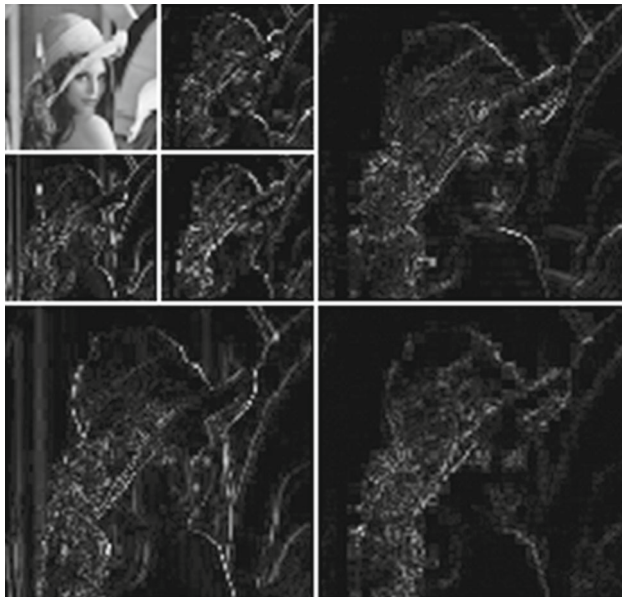


Fig. 3 Image decomposed into 2 levels

old values of the image are determined based on the reduced histogram of the image. The reduction in the length of the histogram enables the proposed methods to run faster. Finally, the threshold values determined are mapped to the original image space and are used to segment the image.

3.2 Combined Objective Function

Selection of a proper threshold is important for image segmentation problems. The traditional Otsu method (1978) provides better segmentation results for an ordinary image. It performs well if the image has bimodal histogram dis-

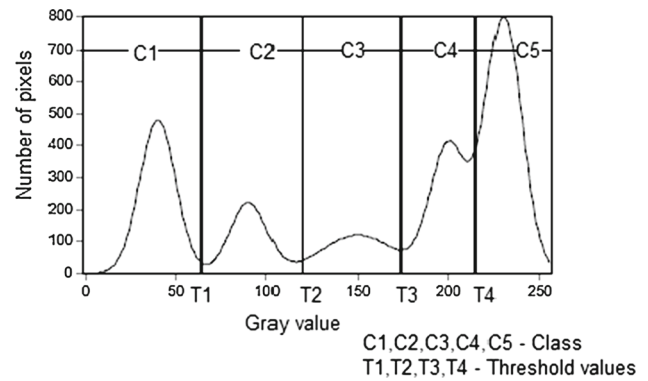


Fig. 4 Histogram plot

tribution. When there is a distinct difference between the foreground and background objects, then Otsu method gives better results. If the image boundaries are overlapped with each other, then it is difficult to obtain a proper threshold value for them. The complex image has more than one maxima. The computed maxima may be a local maxima which gives poor segmentation results. Hence for complex images, it does not give good results. This paper uses Combined Objective Function (COF) to determine optimal number of thresholds and their values for the given image using GA, PSO and hybrid techniques. All these optimization techniques use the same proposed Combined Objective Function to solve the problem and they lead to a feasible solution. The binary strings of 0s and 1s are generated from the reduced histogram and it is used as an input to the problem. The histogram plot in Fig. 4 contains peaks and valleys of the intensity values. It is inferred from Fig. 4 that there are four optimal thresholds ($v - 1$) present and their values are T1, T2, T3, T4 positions, respectively. The ($v - 1$) thresholds segment the feature space into v classes. The ranges for the classes are listed as follows:

1. Class 1 (C1) falls in the range between 0 to T1.
2. Class 2 (C2) falls in the range between T1 to T2.
3. Class 3 (C3) falls in the range between T2 to T3.
4. Class 4 (C4) falls in the range between T3 to T4.
5. Class 5 (C5) falls in the range between T4 to 255.

For the proposed methods, to determine multiple thresholds and values, a new COF is coined by using the following criteria:

1. Threshold criterion
2. Within-class variance
3. Between-class variance
4. Goodness
5. Selection.

There is no single criterion used for the segmentation problem for all types of images. To have better segmentation for all type of images, more criteria are to be considered (Sridevi et al. 2014).

3.2.1 Threshold criterion

The number of threshold is determined from the binary string by counting number of zero bits present in it. The position occupied by those zero bits indicate the threshold value for the given image. Equation (1) represents number of bits required to represent the thresholded image.

$$f1(v) = \min(\log_2(v^2)), \quad (1)$$

where v is optimal class number.

The thresholded classes should be distinctly related to their pixel intensity values. The increase in number of classes will

1. decrease the difference between the original and thresholded image and
2. increase total number of bits required to represent the thresholded image.

3.2.2 Within-class variance criterion

The variance within the classes should be minimum. It provides discrepancy between the original and thresholded image. The within-class variance is computed based on two parameters namely, mean gray value of each class and normalized probability. The objective function for within-class variance as given in Eq. 2 is a minimization function which in turn minimizes the mean gray value of each class and normalized probability parameters.

$$f2(v) = \min(\sigma_w^2(v))^{1/2} \quad (2)$$

3.2.3 Between-class variance criterion

The between-class variance criterion is a measure of separability between classes (Gonzalez and Woods 2008). It gives optimal global threshold value. The method mainly depends on the computed histogram of the image such as probability of class, mean gray value of each class, normalized histogram, total mean intensity value of the image. The objective of the between-class variance is mentioned in Eq. 3 which is a maximization function as discussed in (Otsu 1978).

$$f3(v) = \max(\sigma_B^2(v)) \quad (3)$$

3.2.4 Goodness criterion

The goodness is a metric used along with between classes to measure the separability between them (Gonzalez and Woods 2008). The use of this criterion is to find out the threshold value by maximizing the objective function using variance between the class and total variance of the image. The objective function of goodness criteria is evaluated using Eq. 4.

$$f4(v) = \max\left(\frac{\sigma_B^2(v)}{\sigma_T^2}\right) \quad (4)$$

3.2.5 Selection criterion

The quality of selected threshold image can be evaluated using uniformity measure. It is computed based on the following parameters

1. Number of classes (v),
2. Gray value (f_i) of the pixel position i ,
3. Maximum (f_{\max}) and minimum (f_{\min}) gray values of the image and
4. Gray level mean value (m_i) of the pixels in the class j .

If the uniformity is high, then the quality of the thresholded image is better. Hence the selection criteria based on uniformity is a maximization function. The fitness function for determining optimal threshold using uniformity measures (Gonzalez and Woods 2008) is given in Eq. 5.

$$f5(v) = \max\left[1 - (2 \times v) \times \left(\frac{\sum_{j=0}^v \sum_{i \in C_j} (f_i - m_i)^2}{M \times N \times (f_{\max} - f_{\min})^2}\right)\right], \quad (5)$$

where c_j is segmented region within the class j .

The proposed method uses a newly coined objective function which is derived from Eqs. 1–5. The proposed objective function as given in Eq. 6 is a minimization function which determines a set of optimal threshold values ($t_1, t_2 \dots t_{v-1}$).

$$f(v) = \min[f1(v) + f2(v) - f3(v) - f4(v) - f5(v)] \quad (6)$$

The objective of the threshold segmentation problem is to select proper number of threshold and values. If the class number is high, then the thresholded image is similar to the original image. Minimal objective function value gives better segmentation result. The coined objective function uses uniformity, variance, separability, discrepancy properties of the image. Hence, it gives a minimal fitness value.

3.3 Sequential version

This section uses optimization techniques such as GA and PSO to determine the optimal threshold values in a robust manner. The reduced binary string is passed to the optimizer (either GA or PSO) to get the best threshold. The optimizer uses minimization objective function as discussed in Sect. 3.2. The objective is to group the pixels of the image into different groups based on the threshold values. The following Sects. 3.3.1 and 3.3.2 elaborate the working of standard GA and PSO techniques for OTS. The threshold is determined from the peak and the valley of the histogram. If the value of the string is 1, then it denotes the peak, else it is a valley. Total number of zero bits present in the string indicates the number of threshold values.

3.3.1 OTS using SGA

The reduced histogram is obtained as a result of DWT using j levels, which are given as an input (binary string) to GA optimizer to find best threshold and appropriate values. The same learning strategy used by Hammouche et al. (2008) is utilized here to find an optimal solution. GA operation (Goldberg 1989) such as selection, cross-over and mutation are performed. The fitness value is computed using Eq. 6. The mating pool is filled by selecting any two strings randomly from the current population. The string which has the best fitness value is placed in the mating pool. Any two strings are chosen from the mating pool to produce two offsprings by exchanging all the bits between them. The cross-over operation takes place if the random value is lower than the cross-over probability. The result of cross-over may lead to a string of zero bits. They do not form distinct range for the multiple threshold values and it gives continuous bits as threshold values. To avoid this situation, mutation is performed by changing the bit string values. The best fitness value with the string in each generation is determined and the best value is carried over to the next generation. GA terminates with the best fitness value, when convergence occurs in the fitness value or maximum number of generations is reached. The algorithmic steps for the proposed OTS using SGA are given in Algorithm 1.

Algorithm 1: OTS using SGA

Input to the algorithm: Histogram of reduced image.

Output of the algorithm: Optimal number of thresholds and their values

Termination criteria: Either number of generations or error factor.

The steps involved in the algorithm are given below:

1. Generate initial population.

2. Compute fitness value for each population using Eq. 6 and store the best fitness value along with its string (S^P).
3. Perform GA operations such as selection, cross-over and mutation to generate next population.
4. Determine best fitness value by comparing the best string of current population (S^C) with best string (S^P) in previous population.
If the fitness value of string $S^C < S^P$, then replace S^P with S^C .
5. Execute steps 2 to 4 until a termination criterion is reached.

Algorithm 2: OTS using SPSO

Input to the algorithm: Histogram of reduced image.

Output of the algorithm: Optimal number of thresholds and their values

Termination criteria: Either number of generations or error factor.

The steps involved in the algorithm are given below:

1. Generate initial position of particles.
2. Compute fitness value for each particle using Eq. 6 and store the best fitness value along with its string (S^P).
3. Perform PSO operations such as updation of velocity and position using Eqs. 7 and 8.
4. Determine best fitness value by comparing the best string of current particle (S^C) with best string (S^P) in previous particle.
If the fitness value of string $S^C < S^P$, then replace S^P with S^C .
5. Execute steps 2 to 4 until a termination criterion is reached.

3.3.2 OTS using SPSO

Binary PSO is applied to solve the proposed COF, as the input used for the PSO is a binary string of 0s and 1s. The particles are updated based on the position and the velocity. The velocity is interpreted as a probability to interchange the values of 0 and 1 while updating the particle position by using Eq. 9. The particle velocity and position http://en.wikipedia.org/wiki/Particle_swarm_optimization are updated using Eqs. 7 and 8, respectively.

$$\begin{aligned} \text{vel}_{is}(t+1) = & \varphi \times x_{is}(t) + c_1 r_1 (y_{is}(t) - x_{is}(t)) \\ & + c_2 r_2 (y_{js}(t) - x_{is}(t)) \end{aligned} \quad (7)$$

$$x_{is}(t+1) = \begin{cases} 0 & \text{if } \text{sig}(\text{vel}_{is}(t+1)) \leq r \\ 1 & \text{if } \text{sig}(\text{vel}_{is}(t+1)) > r \end{cases}, \quad (8)$$

where

$$\text{sig}(x) = \frac{1}{1 + e^{-x}} \quad (9)$$

$1 \leq i \leq np$ and $1 \leq s \leq S$; $x_{is}(t)$ is the current position of the particle; $vel_{is}(t)$ is the current velocity of the particle; $y_{is}(t)$ is the personal best position of the particles; $yy_{js}(t)$ is the global best position of the swarm; r , r_1 and r_2 are the random numbers within $(0, 1)$; c_1 and c_2 are accelerating constants; φ denotes inertia weight to control the previous velocity impact; S is the search space; np is number of particles

The number of particles, their positions and velocity are initialized in a random manner. Each particle evaluates the proposed COF and finds the local best of each particle and computes the global best among all particles. The positions and velocity are updated using Eqs. 8 and 7 based on the best value. The process is repeated until the terminating criteria are reached. Algorithm 2 depicts the algorithmic steps of the proposed OTS using SPSO.

3.4 Parallel version

Though the GA and PSO techniques give an optimal solution, the time taken for convergence is more. To reduce the time, proposed methods using GA and PSO are executed in a parallel environment. The following architectures are available to execute the algorithms in a parallel environment using shared memory (Akl 1990).

1. EREW SIMD, 2. ERCW SIMD, 3. CREW SIMD, 4. CRCW SIMD, 5. EREW MISD, 6. ERCW MIMD, 7. CREW MIMD, 8. CRCW MIMD, 9. EREW MIMD, 10. ERCW MIMD, 11. CREW MIMD, 12. CRCW MIMD

For this work, Exclusive Read Exclusive Write Shared Memory Single Instruction Multiple Data (EREW SM SIMD) parallel architecture is used. The fitness function of the population is computed in a parallel manner using K processors. Each processor reads the input string from the shared memory and evaluates the same fitness function (single instruction) for different population (different data) in the local memory of the respective processor. The computed minimal fitness value for the current iteration is written back to shared memory and updated based on the previous and currently computed values.

3.4.1 OTS using PGA

The method discussed in Sect. 3.3.1 is executed in a parallel environment to reduce the execution time. EREW SM SIMD model is used to create the parallel environment with K number of processors where each processor should be assigned with 1 or 2 strings from the population size. For this work, it has been assumed that the number of processors (K) is equal to the population size. The binary strings are read by each processor simultaneously from the SM and the fitness value is evaluated in the local memory of each processor for the

corresponding population. The best fitness value among all is found out and written in SM and it is used for the next generation. The parallelism makes the algorithm to execute faster. The steps involved in PGA are given in Algorithm 3.

3.4.2 OTS using PPSO

To reduce the computational time of the proposed OTS using PSO algorithm, EREW SM SIMD parallel architecture with K number of processors is used. Assume that the number of processors used is equal to the number of particles initialized. Each and every processor evaluates the objective function in a parallel manner and the local best of each processor is written back to the SM. The global best value is found among local best values. The process is repeated until the terminating criteria are reached. Algorithm 4 gives the steps involved in the proposed PPSO algorithm.

Algorithm 3: OTS using PGA

Input to the algorithm: Histogram of reduced image.

Output of the algorithm: Optimal number of thresholds and their values

Termination criteria: Either number of generations or error factor.

The steps involved in the algorithm are given below:

1. Generate initial population.
2. do in parallel {for each population}
 - a) Compute fitness value using Eq. 6 and store the best fitness along with its string (S^P).
 - b) Perform GA operations such as selection, cross-over and mutation to generate next population.
 - c) Determine best fitness value by comparing the best string of current population (S^C) with best string (S^P) in previous population.
If the fitness value of string $S^C < S^P$, then replace S^P with S^C .
3. Execute steps 2 a) to c) until a termination criterion is reached.

Algorithm 4: OTS using PPSO

Input to the algorithm: Histogram of reduced image.

Output of the algorithm: Optimal number of thresholds and their values

Termination criteria: Either number of generations or error factor.

The steps involved in the algorithm are given below:

1. Generate initial position of particle.
2. do in parallel {for each particle}
 - a) Compute fitness value using Eq. 6 and store the best fitness along with its string (S^P).

- b) Perform PSO operations such as updation of velocity and position using Eqs. 7 and 8, respectively.
- c) Determine best fitness value by comparing the best string of current particle (S^C) with best string (S^P) in previous population.
If the fitness value of string $S^C < S^P$, then replace S^P with S^C .
3. Execute steps 2 a) to c) until a termination criterion is reached.

3.5 Hybrid method

The fitness value of GA and PSO can be further refined using hybrid model. This paper uses a hybrid optimization function using pattern search algorithm. The evaluation of objective function using pattern search alone can trap on local minima based on the initial random value chosen by the algorithm. To avoid local minima and to get refined fitness value, pattern search algorithm (Goldberg 1989) can be combined with optimization techniques such as GA, PSO or simulated annealing, etc. The pattern search finds minimum value for the given objective function. In the proposed hybrid methods, the pattern search is combined with GA or PSO to get better threshold values for segmentation problem. The solver first runs GA or PSO and finds out the best fitness value for it. This result is given as an input (initial value) to the pattern search algorithm. The hybrid models provide better solution to the optimization problem. The algorithmic steps of the proposed HGA and HPSO algorithms are given in Algorithms 5 and 6, respectively.

Even though the proposed hybrid model gives better results when compared to standard GA or PSO, it takes more time for executing the algorithm. To reduce the computational time of the hybrid model further, the proposed hybrid model is executed in a parallel environment using EREW SM SIMD parallel architecture. Parallelism is attained in GA or PSO technique by evaluating the fitness function in a parallel manner as explained in the Sects. 3.4.1 and 3.4.2. The resultant value of parallel GA or PSO is then given to the pattern search algorithm to get better fitness and threshold values. Algorithms 7 and 8 give the algorithmic steps of the HPGA and HPPSO.

Algorithm 5: OTS using HGA

Input to the algorithm: Histogram of reduced image.

Output of the algorithm: Optimal number of thresholds and their values

Termination criteria: Either number of generations or error factor.

The steps involved in the algorithm are given below:

1. Generate initial population.

2. Compute fitness value for each population using Eq. 6 and store the best fitness along with its string (S^P).
3. Perform GA operations such as selection, cross-over and mutation to generate next population.
4. Determine best fitness value by comparing the best string of current population (S^C) with best string (S^P) in previous population.

If the fitness value of string $S^C < S^P$, then replace S^P with S^C .

5. Execute steps 2 to 4 until a termination criterion is reached.
6. Result of GA is given as an input to pattern search algorithm.
7. Determine best fitness value for the hybrid model.

Algorithm 6: OTS using HPSO

Input to the algorithm: Histogram of reduced image.

Output of the algorithm: Optimal number of thresholds and their values

Termination criteria: Either number of generations or error factor.

The steps involved in the algorithm are given below:

1. Generate initial position of particles.
2. Compute fitness value for each particle using Eq. 6 and store the best fitness along with its string (S^P).
3. Perform PSO operations such as updation of velocity and position using Eqs. 7 and 8, respectively.
4. Determine best fitness value by comparing the best string of current particle (S^C) with best string (S^P) in previous particle.

If the fitness value of string $S^C < S^P$, then replace S^P with S^C .

5. Execute steps 2 to 4 until a termination criterion is reached.
6. Result of PSO is given as an input to pattern search algorithm.
7. Determine best fitness value for the hybrid model.

Algorithm 7: OTS using HPGA

Input to the algorithm: Histogram of reduced image.

Output of the algorithm: Optimal number of thresholds and their values

Termination criteria: Either number of generations or error factor.

The steps involved in the algorithm are given below:

1. Generate initial populations.
2. do in parallel {for each population}

- a) Compute fitness value using Eq. 6 and store the best fitness along with its string (S^P).
 - b) Perform GA operations such as selection, cross-over and mutation to generate next population.
 - c) Determine best fitness value by comparing the best string of current population (S^C) with best string (S^P) in previous population.
If the fitness value of string $S^C < S^P$, then replace S^P with S^C .
 - d) Execute steps 2 a) to c) until a termination criterion is reached.
3. Result of GA is given as an input to pattern search algorithm.
 4. Determine best fitness value for the hybrid model.

Algorithm 8: OTS using HPPSO

Input to the algorithm: Histogram of reduced image.

Output of the algorithm: Optimal number of thresholds and their values

Termination criteria: Either number of generations or error factor.

The steps involved in the algorithm are given below:

1. Generate initial position of particles.
2. do in parallel {for each particle}
 - a) Compute fitness value using Eq. 6 and store the best fitness along with its string (S^P).
 - b) Perform PSO operations such as updation of velocity and position using Eqs. 7 and 8, respectively.
 - c) Determine best fitness value by comparing the best string of current particle (S^C) with best string (S^P) in previous particle
If the fitness value of string $S^C < S^P$, then replace S^P with S^C .
 - d) Execute steps 2 a) to c) until a termination criterion is reached.
3. Result of PSO is given as an input to pattern search algorithm.
4. Determine best fitness value for the hybrid model.

3.6 Determination of optimal threshold values

Threshold can be determined from the zero bits in the string which correspond to the best fitness. Threshold values are in the reduced image space. Hence, they have to be converted to the original image space (Sridevi et al. 2014) by using Eq. 10.

$$t_o = t_i \times 2^j, \tag{10}$$

where t_o gives threshold value in original image space; t_i is the threshold value in reduced image space, for $1 \leq i \leq v - 1$ and j is the decomposition level.

The determined threshold values may get changed due to stochastic nature of the optimizer. To avoid this, iterative refinement process (Hammouche et al. 2008) is applied to get accurate threshold values.

Example Let the input string be 1,1,0,1,1,1,1,0,1,1,1,1,1,0,1,1. Number of zero bits indicate threshold number and their corresponding position denotes the threshold values. In this example, 3 zero bits are present which indicates the number of thresholds and their values are in the bit positions 3, 8 and 14, respectively. The obtained thresholds are in the reduced image space at the decomposition level 2. The obtained values are mapped to the original image space by using Eq. 10. The determined threshold values are 12, 32 and 56 in the original space for the input string. The input gray image is segmented based on the determined threshold values.

3.7 Image segmentation

The original image $f(x, y)$ is segmented into v sub-regions. The segmented image $g(x, y)$ (Gonzalez and Woods 2008) can be obtained from the determined threshold values using Eq. 11.

$$g(x, y) = \left\{ \begin{array}{ll} a & \text{if } 0 < f(x, y) \leq t_1 \\ b & \text{if } t_1 < f(x, y) \leq t_2 \\ \vdots & \vdots \\ c & \text{if } t_{v-1} < f(x, y) \leq 255 \end{array} \right\}, \tag{11}$$

where a, b and c are intensity values between 0 to 255; $t_i (i = 1 \text{ to } v - 1)$ is the threshold value.

The proposed methods give optimal class number and threshold values. The resultant image is used for further processing such as feature extraction, classification, recognition, analysis and understanding, etc. For the real-time applications, the proposed OTS has to be run faster by reducing the input image using DWT and by using a parallel environment.

4 Results and performance analysis

The proposed multilevel thresholding methods are implemented in MATLAB 7.11 for different benchmark images (<http://www.eecs.berkeley.edu/Research>). The input image shown in Fig. 5a is used for performance evaluation of the proposed method. The histogram plot for the input image is shown in Fig. 5b. From the histogram plot, it is inferred that the intensity distribution is not distinct for the given image. Hence, it requires multiple thresholds to segment the image. To get optimal values for these multiple thresholds, proposed methods use the following optimization techniques for execution in both sequential and parallel environment:

1. OTS using GA
2. OTS using PSO
3. OTS using HGA
4. OTS using HPSO.

4.1 Parameters

The parameters of GA and PSO methods are initialized to execute the proposed methods. They are derived using conventional methods found in the literature and their usage follows the standard procedure. The parameters used in GA and PSO are given in Sects. 4.1.1 and 4.1.2, respectively.

4.1.1 GA parameters

The parameters used for GA are listed as follows:

1. Tournament selection process is used in the selection operation of GA
2. Cross-over probability is 0.8
3. Mutation probability is 0.2
4. Population size and number of generations considered are 20 and 100, respectively.

4.1.2 PSO parameters

The following PSO parameters are initialized to execute the proposed algorithms:

1. Number of particles considered is 20
2. Number of generations considered is 100
3. Accelerating constant is 1.25
4. Inertia weight is 0.5.

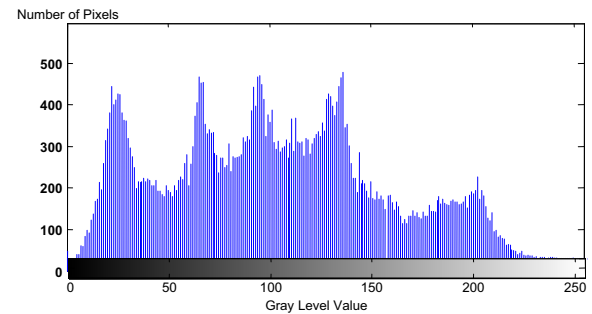
4.2 Experimental results

The experiment was carried out with (<http://www.eecs.berkeley.edu/Research>) different benchmark images to test the proposed methods. The proposed methods are executed with parameters mentioned in Sects. 4.1.1 and 4.1.2 for 50 runs. The segmentation results are shown in Figs. 5, 6 and 7 for Lena image, medical image and an aeroplane image, respectively.

The famous benchmark Lena image is used to test the proposed methods namely, GA, PSO, HGA and HPSO. Figure 5b indicates the histogram plot for the input image shown in Fig. 5a. The results using GA, PSO, HGA and HPSO are shown in Fig. 5c, f, respectively. Similarly, the results for the two different input images (medical and video frame) are shown in Figs. 6 and 7. The segmented image is used for further processing. In case of medical images, it is used for classification and diagnosis of diseases. Figure 7 shows an aeroplane image which is a frame of a video sequence. The



(a) Input image



(b) Histogram plot



(c) GA



(d) PSO



(e) HGA



(f) HPSO

Fig. 5 Results of segmentation of a Lena image

resultant image is used to track the flying aeroplane. The execution time of the proposed model is reduced using parallel versions of the algorithms in order to suit the real-time applications. From the results, it is inferred that HPSO gives optimal threshold.

The experiments are carried out for a noisy input image shown in Fig. 8a and their resultant thresholded images are shown in Fig. 8b, e. From the results, it is inferred that the proposed algorithms perform well for noisy images also.

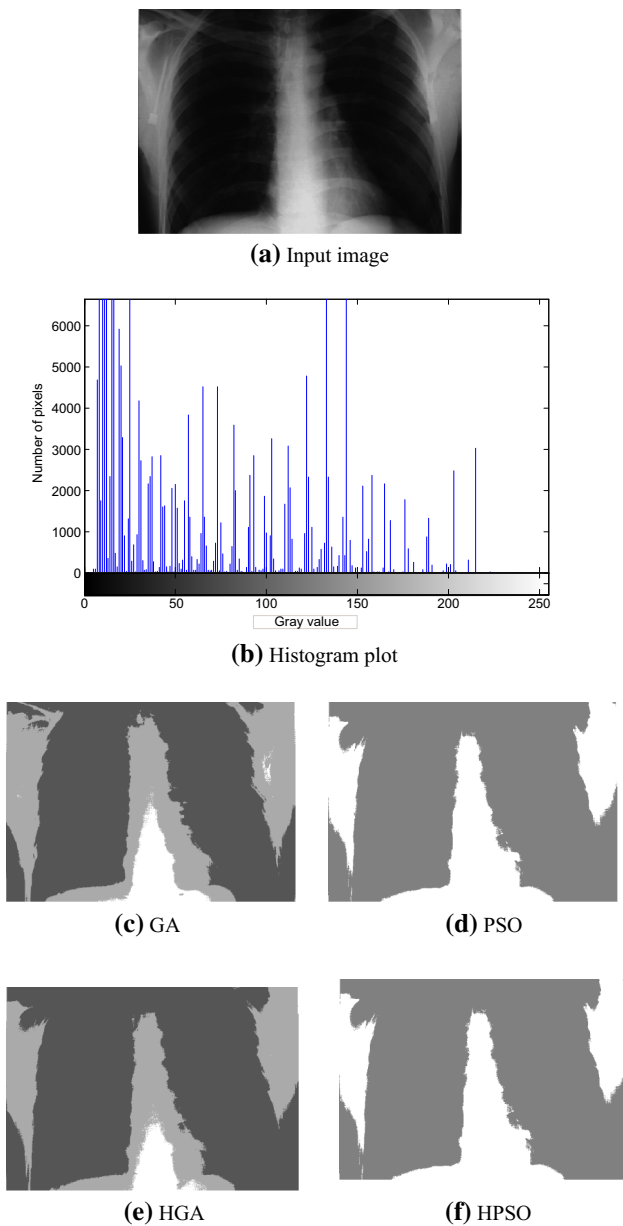


Fig. 6 Results of segmentation of a medical image

4.3 Performance analysis

From the experimental results of the proposed method shown in Sect. 4.2, it is inferred that the methods can be applied to any kind of image applications such as medical, object or video tracking, recognition, retrieval, etc. The performance of the proposed method is compared with the existing methods such as Kapur, Otsu, COF using ACT and goodness criteria (Sridevi et al. 2014). As discussed in Sect. 3.2, the fitness function is a minimization function. The obtained fitness value for GA and PSO is shown in the Figs. 9 and 10, respectively. From the plot, it is inferred that the proposed methods using GA, PSO, PGA and PPSO gives minimum

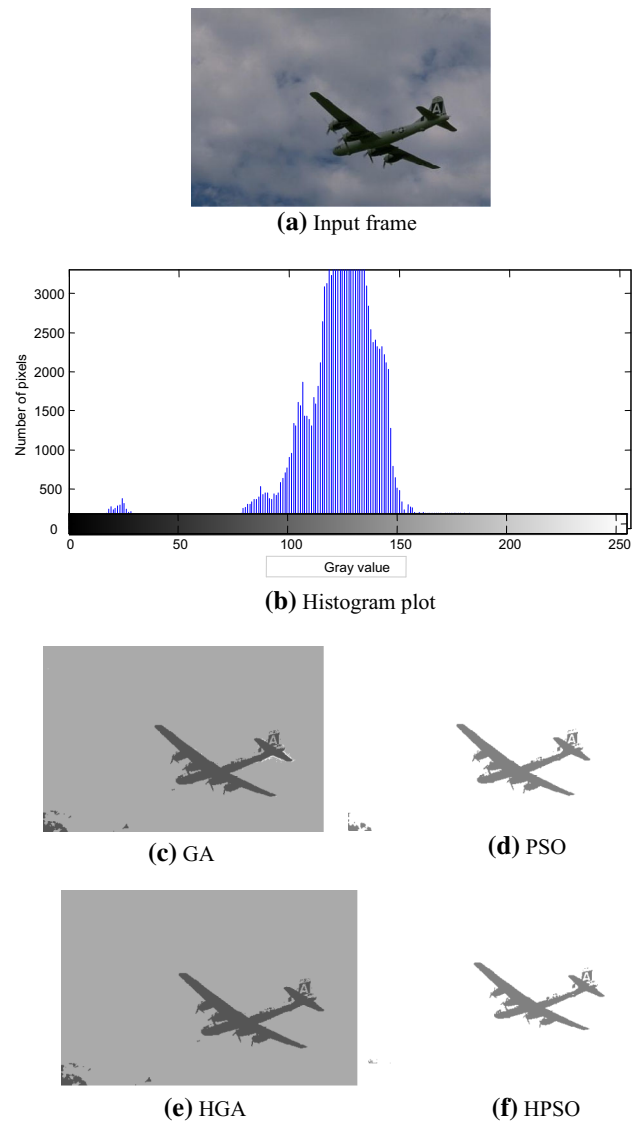


Fig. 7 Results of segmentation for a video frame

fitness value when compared to the existing methods. The proposed methods combine five different image segmentation criteria to determine the optimal threshold value. Hence, the fitness value of OTS using the proposed new COF is the minimum and hence gives optimal number of classes and thresholds.

The execution time of proposed methods is compared with existing methods such as Kapur, Otsu, ACT and goodness COF (Sridevi et al. 2014) methods using both GA and PSO techniques. Figure 11 depicts the execution time of the proposed and existing methods. It is inferred from the graph that the execution time of the proposed methods are less than the existing methods.

To suit the proposed methods to work on real-time applications such as object tracking and recognition in a video, image classification, image matching, biometric, telemedicine, fea-

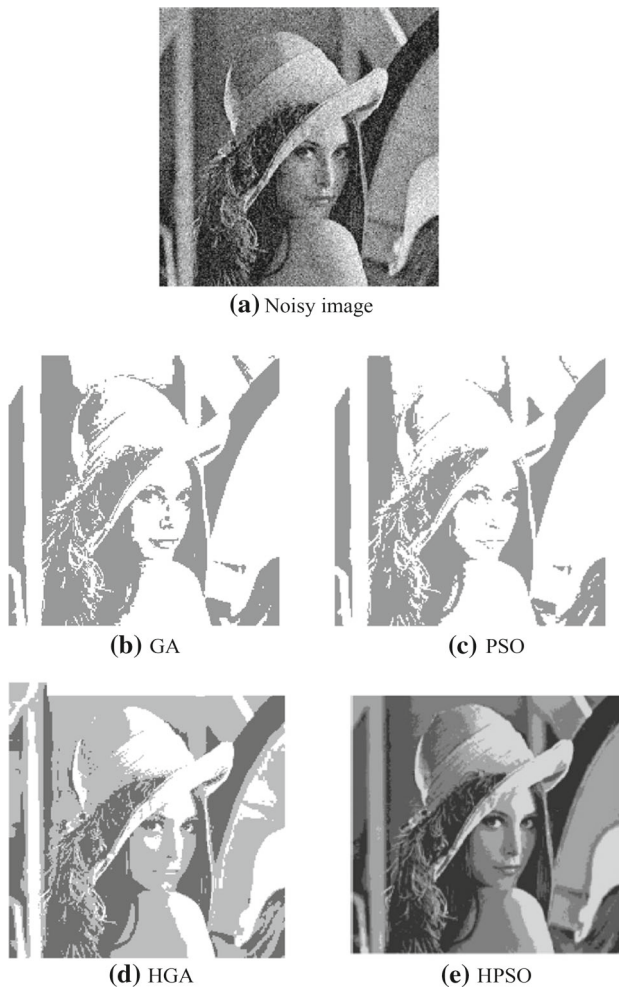
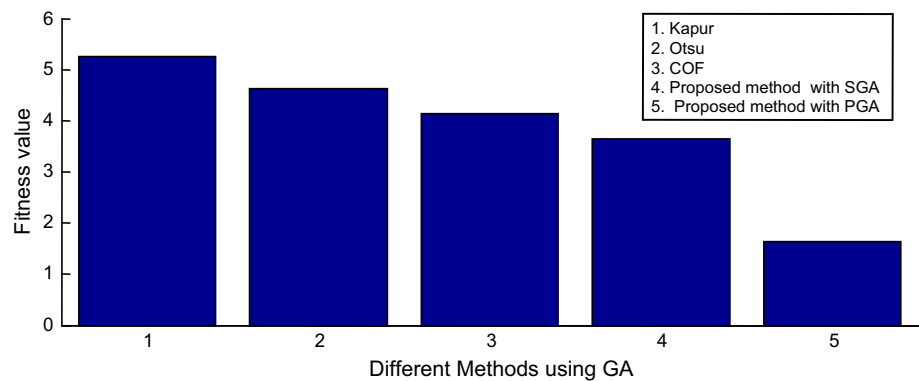


Fig. 8 Resultant thresholded image for the noisy image

ture extraction, image retrieval, etc., and the execution time of the algorithms is to be reduced. This can be done by using parallel processors instead of using uniprocessor. The execution time of the proposed algorithm is compared with GA technique using uniprocessor and multiprocessor as shown in the Fig. 12. The proposed OTS using PSO is executed in a sequential and parallel environment and the results are

Fig. 9 Fitness value vs OTS methods (using GA)



shown in Fig. 13. The execution time of the parallel version of proposed OTS using GA and PSO techniques are compared and the plot is shown Fig. 14 by varying the number of processors. From the plot, it is inferred that the execution time is inversely proportional to the number of processors used in both PGA and PPSO.

The fitness function of the proposed method can be further minimized by using a hybrid optimization technique. The result of GA is given as the input for pattern search algorithm to get better fitness and threshold values. Similarly the PSO technique can be made hybrid with the pattern search algorithm. The fitness value and execution time of the hybrid models (HGA and HPSO) are compared with the standard GA and PSO. Based on the values obtained, a graph is plotted and it is shown in Fig. 15. It is inferred that the fitness value of the hybrid models are less than the standard techniques which indicate that the fitness value is well refined and produces optimal threshold values. But the execution time in hybrid model is more, as the objective function is computed twice. To reduce the computational time of hybrid methods, they are executed in a parallel environment using EREW SM SIMD architecture. The fitness and execution time of proposed hybrid parallel methods (HPGA and HPPSO) are compared with the parallel GA and PSO methods by keeping number of processors ($K = 8$) as constant. From the Fig. 16, it is inferred that the execution time is drastically reduced in parallel versions of hybrid models. Also segmentation performed using HPPSO gives proper homogeneous sub-regions which can be seen from Figs. 5f, 6f and 7f for the input Lena, X-ray and aeroplane image, respectively.

The execution time of the proposed methods namely, GA, PSO, HGA and HPSO are compared among themselves in both sequential and parallel environments. The graph is plotted for the results and it is shown in Fig. 17. The hybrid methods take more time than sequential and parallel versions of GA and PSO methods. But the parallel versions of the proposed methods take less time than GA, PSO and hybrid GA and PSO methods. Hence, the proposed HPGA and HPPSO are well suited for real-time applications to get results faster.

Fig. 10 Fitness value vs OTS methods (using PSO)

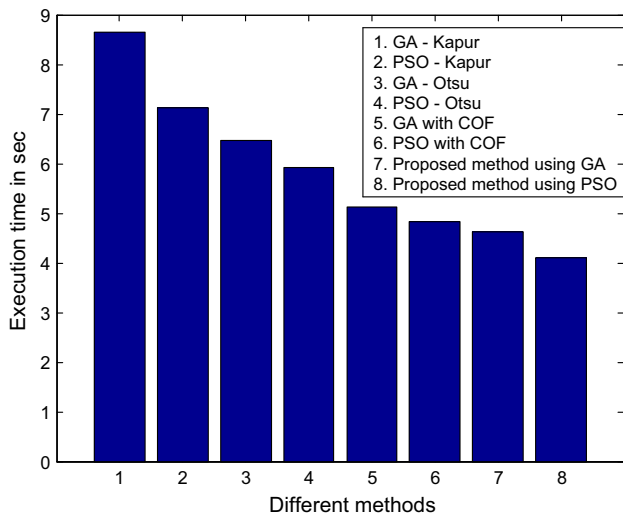
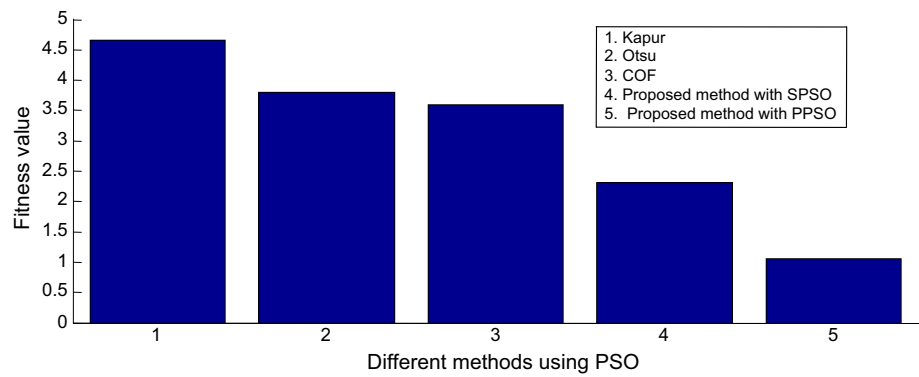
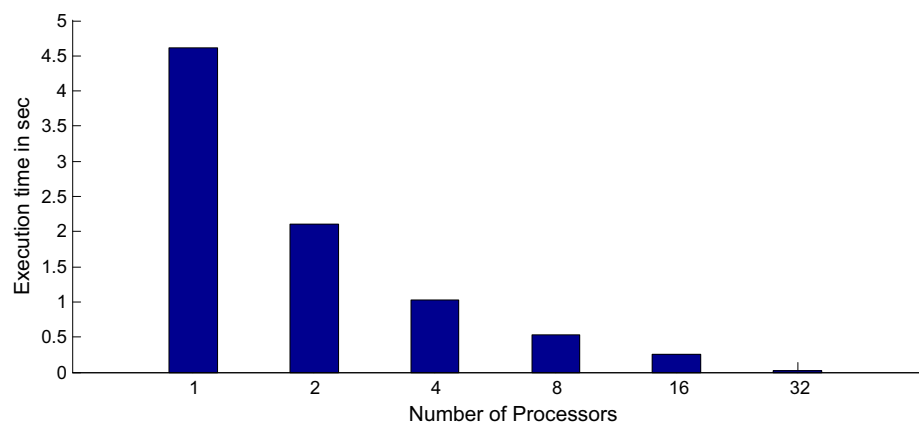


Fig. 11 Execution time vs OTS methods

The determined optimal threshold value mainly depends on the fitness value of the function. The two parameters namely, number of generations and population size, affect the fitness value of the function. The experiments were conducted by varying the above two parameters for the proposed OTS methods to run in sequential and parallel manner. The graph is plotted from the obtained experimental results and

Fig. 12 Execution time of GA vs Number of processors



they are shown in Figs. 18 and 19. In Fig. 18, fitness value is evaluated by varying number of generations from 10 to 100. The fitness value of proposed methods is compared with existing methods (Kapur, Otsu). From the plot, it is inferred that the fitness value decreases with increase in number of generations in all methods. The analysis is made based on the fitness value and various population size (10 to 100). The graph is plotted for the fitness value with respect to the population size and it is shown in Fig. 19. It is inferred that when population size increases, then the fitness value of the method decreases. The fitness value of hybrid parallel PSO is the minimal among all the proposed methods and so it gives best set of optimal thresholds and segments for the given image.

4.4 Timing analysis

This section analyzes the execution time of the proposed methods in uniprocessor and multiprocessors. The proposed OTS methods using GA, PSO and hybrid GA and PSO is executed in a single processor. The time taken for executing those algorithms for the input image shown in Fig. 5a is tabulated in Table 1. And the same is executed in the parallel environment by varying the number of processors (2, 4, 8, 16, 32) to attain parallelism. These results are tabulated in

Fig. 13 Execution time of PSO vs Number of processors

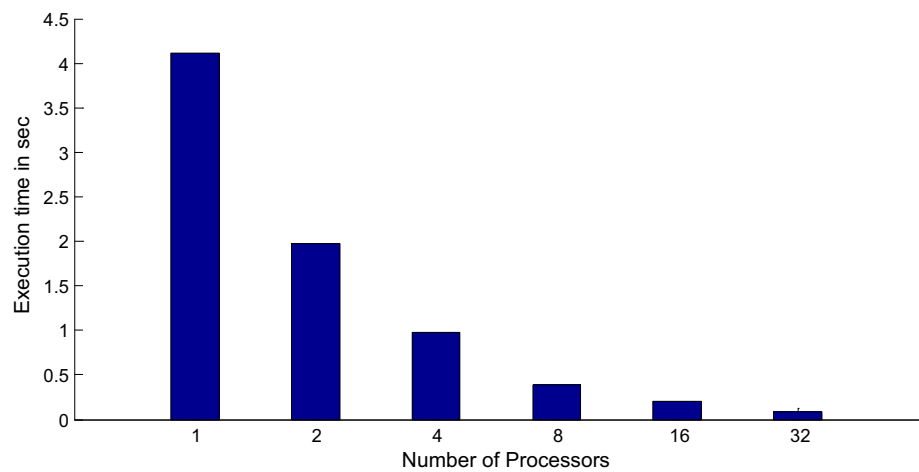


Fig. 14 Execution time of GA and PSO vs Number of processors

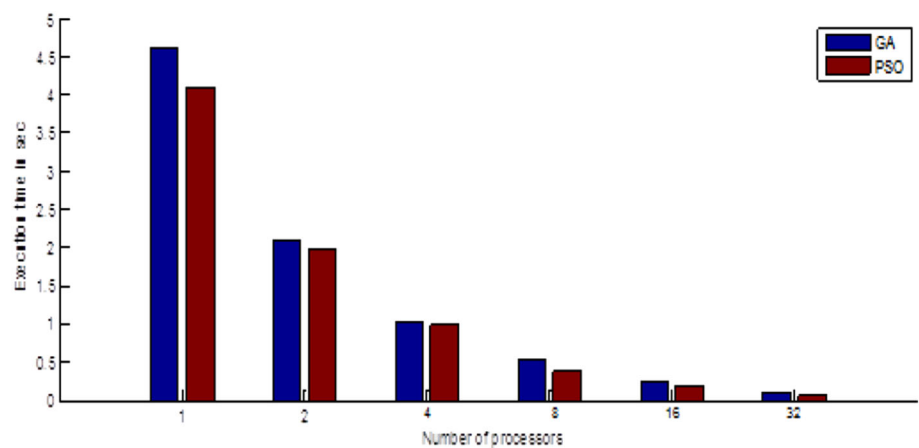


Fig. 15 Fitness value, execution time vs Soft Computing techniques

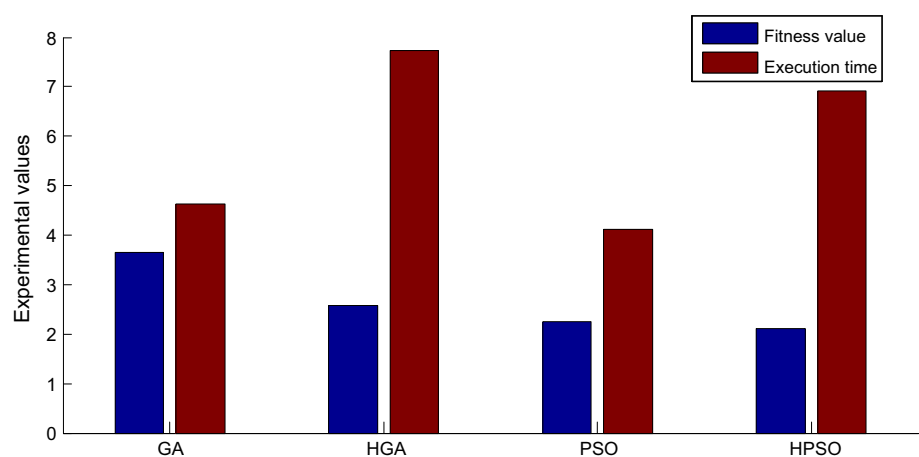


Table 2. From the results, it is analyzed that the sequential PSO takes less time than GA. The hybrid GA and PSO methods take more time to get optimal threshold values as they perform the optimization twice.

Table 3 depicts the execution time of proposed OTS methods in sequential and parallel manner. Eight processors are used for executing OTS methods in a parallel environment.

The multilevel threshold value mainly depends on the number of threshold and their values. In the proposed methods, the optimal thresholds are determined by minimizing the objective function. The proposed methods are executed for various population sizes in sequential and parallel environment. The results (fitness value and execution time) are tabulated in Tables 4 and 5 for GA, PSO and hybrid techniques. From the tables, it is clear that the

Fig. 16 Fitness values, execution time vs Soft Computing techniques ($K = 8$)

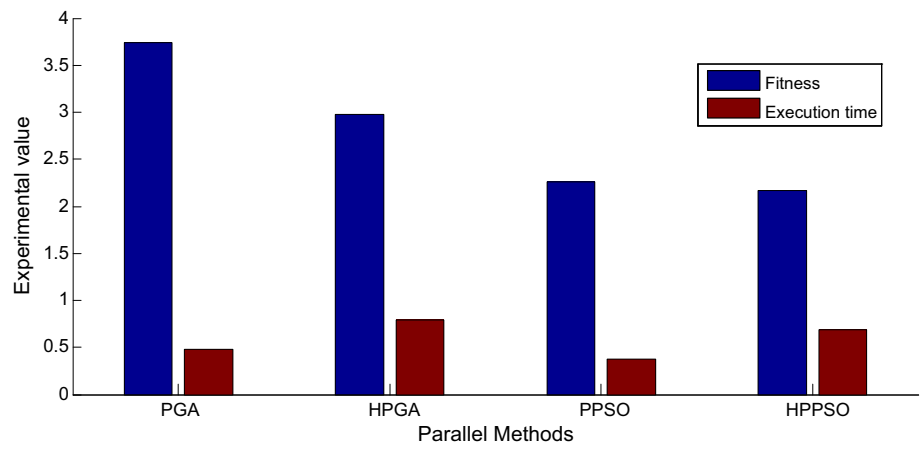


Fig. 17 Execution time vs Soft Computing techniques

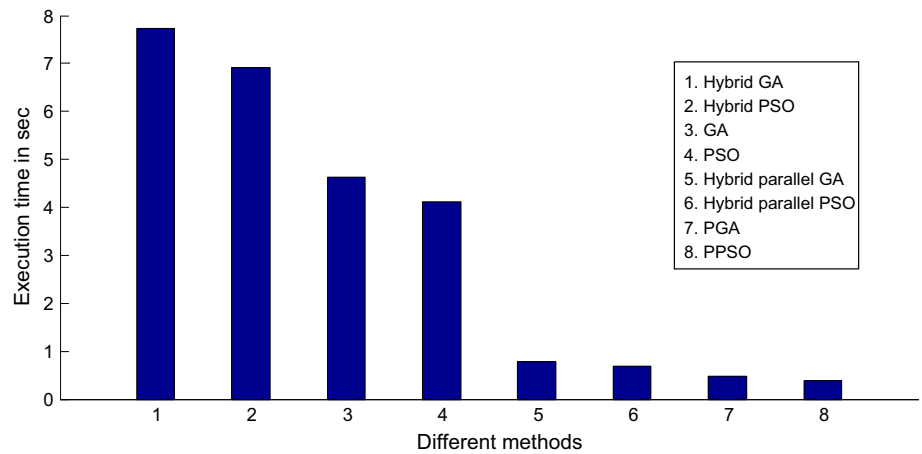


Fig. 18 Fitness value vs Number of Generations

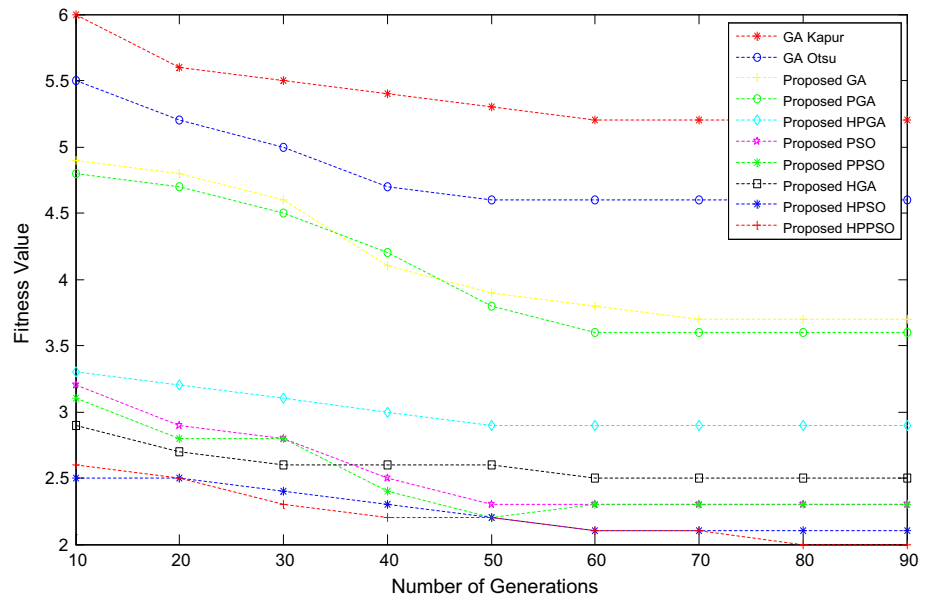


Fig. 19 Fitness value vs Population size

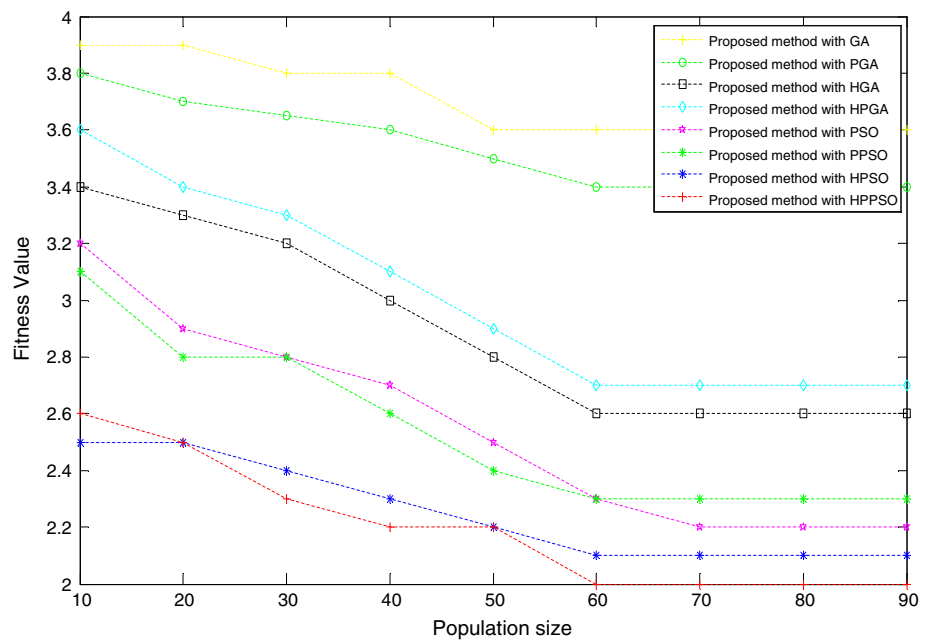


Table 1 Execution time of proposed OTS methods in uniprocessor

OTS using	Execution time (s)
GA	4.62
PSO	4.11
HGA	7.73
HPSO	6.91

Table 2 Execution time of proposed OTS methods in parallel environment

Number of processors (<i>K</i>)	Execution time of OTS (s)			
	PGA	PPSO	HPGA	HPPSO
2	2.10	1.98	3.61	3.32
4	1.023	0.982	1.69	1.511
8	0.511	0.391	0.789	0.692
16	0.254	0.204	0.399	0.30
32	0.112	0.80	0.23	0.18

1. Fitness value decreases with increase in population size.
2. Execution time increases with increase in population size.

When the population size is increased, then the best optimal thresholds are obtained with minimal fitness value. The determination of optimal thresholds will take more time in sequential environment. To enable the algorithms to work on real-time imaging applications, the proposed methods are executed in parallel environment. The execution time of proposed parallel methods is found out and tabulated in Tables 4 and 5. It is inferred from the tables that the execution time

Table 3 Execution time of proposed OTS methods in sequential and parallel environment

Optimization techniques	Execution time (s)
GA	4.62
GA	4.62
PGA	0.511
HGA	7.73
HPGA	0.789
PSO	4.11
PSO	4.11
PPSO	0.391
HPSO	6.91
HPPSO	0.692

of parallel versions of GA, PSO and hybrid methods is less than their sequential versions.

The objective measures such as fitness value and execution time of the proposed methods are compared with the existing methods such as Otsu and Kapur. The results are tabulated in Table 6. From the analysis, it is observed that the proposed methods take lesser fitness value and execution time. For the input image of Fig. 5a, the resultant threshold values for the proposed methods are shown in Table 7 for 50 runs of execution.

5 Conclusion

In this paper, the multilevel thresholding problem is solved using GA, PSO and hybrid optimization techniques to get

Table 4 Fitness and execution time of proposed OTS methods using GA and PSO for different population size in sequential and parallel environment

Population size	GA				PSO			
	Fitness value		Execution time (s)		Fitness value		Execution time (s)	
	<i>K</i> = 1	<i>K</i> = 8	<i>K</i> = 1	<i>K</i> = 8	<i>K</i> = 1	<i>K</i> = 8	<i>K</i> = 1	<i>K</i> = 8
10	3.9	3.8	2.17	0.20	3.2	3.14	1.87	0.09
20	3.9	3.75	4.62	0.47	3.1	2.9	4.11	0.38
30	3.8	3.6	7.01	0.65	2.89	2.7	6.34	0.54
40	3.6	3.5	9.57	0.92	2.74	2.6	9.01	0.79
50	3.6	3.4	12.03	1.01	2.7	2.6	11.8	0.93

Table 5 Fitness and execution time of proposed OTS methods using hybrid GA and PSO for different population size in sequential and parallel environment

Population size	HGA				HPSO			
	Fitness value		Execution time (in sec)		Fitness value		Execution time (s)	
	<i>K</i> = 1	<i>K</i> = 8	<i>K</i> = 1	<i>K</i> = 8	<i>K</i> = 1	<i>K</i> = 8	<i>K</i> = 1	<i>K</i> = 8
10	3.03	3.14	6.42	0.64	2.5	2.6	5.72	0.53
20	2.89	2.92	7.73	0.789	2.49	2.54	6.91	0.69
30	2.67	2.73	8.03	0.945	2.34	2.41	7.53	0.86
40	2.43	2.52	9.23	1.18	2.25	2.36	8.62	1.01
50	2.41	2.4	10.06	1.30	2.2	2.34	9.67	1.22

Table 6 Comparison of fitness value and execution time of the proposed methods with existing methods

Method	Fitness value	Time (s)
Kapur using GA	5.2562	8.6403
Kapur using PSO	4.6562	7.123
Otsu using GA	4.6260	6.4585
Otsu using PSO	3.797	5.931
OTS using SGA	3.6389	4.62
OTS using SPSO	2.2580	4.11
OTS using PGA	3.743	0.475 (<i>K</i> = 8)
OTS using PPSO	2.2580	0.381 (<i>K</i> = 8)
OTS using HGA	2.5749	7.73
OTS using HPSO	2.1142	6.91
OTS using HPGA	2.975	0.789 (<i>K</i> = 8)
OTS using HPPSO	2.1662	0.692 (<i>K</i> = 8)

Table 7 Optimal threshold values for proposed methods

Method	Threshold values
SGA	34 91 152
SPSO	32 94 147
PGA	31 92 154
PPSO	36 95 158
HGA	42 94 157
HPSO	42 96 152
HPGA	44 95 156
HPPSO	44 95 156

optimal thresholds which are required for real-time imaging applications. The proposed algorithms are implemented in sequential and parallel environments. They are efficient and faster when compared with the existing methods such as Otsu, Kapur methods. It is evident from the experimental results that the proposed HPPSO method gives the best set of optimal threshold values among all other proposed methods. Hence, it is well suited for all types of image and video applications.

References

Akay B (2013) A study on particle swarm optimization and artificial bee colony algorithms for multi level thresholding. *Appl Soft Comput* 13(6):3066–3091

Akl SG (1990) *The design and analysis of parallel algorithms*. Prentice Hall of India, India

Ali M, Ahn CW, Pant M (2014) Multi level image thresholding by synergetic differential evolution. *Appl Soft Comput* 17:1–11 (Elsevier)

Djerou L, Khelil N, Dehimi NH, Batouche M (2012) Automatic multi-level thresholding segmentation based on multi-objective optimization. *J Appl Comput Sci Math* (Suceava) 13(6):25–31

Goldberg DE (1989) *Genetic algorithms: search, optimization and machine learning*. Addison, Wesley

Gonzalez RC, Woods RE (2008) *Digital image processing*. Prentice Hall, New Jersey

Hammouche K, Diaf M, Siarry P (2008) A multilevel automatic thresholding method based on a genetic algorithm for a fast image segmentation. *Comput Vision Image Underst* 109:163–175 (Elsevier)

Indira SU, Ramesh A C (2011) Image segmentation using artificial neural network and genetic algorithm : a comparative analysis.

- In: International conference on Process automation, control and computing, Cimbatore, IEEE Conference Proceedings, pp 1–6
- Kapur JN, Sahoo PK, Wong AKC (1985) A new method for gray-level picture thresholding using entropy of the histogram. *Comput Vision Graph Image Process* 29(3):273–285
- Kaur J, Agrawal S, Vig R (2011) A comparative analysis of thresholding and edge detection segmentation techniques. *Int J Comput Appl* 39:29
- Kim BG, Shim JI, Park DJ (2003) Fast image segmentation based on multi-resolution analysis and wavelets. *Pattern Recognit Lett* 24:2995–3006
- Kumar R, Parashar T, Verma G (2012) Genetic algorithm and DWT based multilevel automatic thresholding approach for vehicle extraction. *Int J Soft Comput Eng (IJSCE)* 2(2):17–21
- Lan J, Zeng Y (2013) Multi-threshold image segmentation using maximum fuzzy entropy based on a new 2D histogram. *Optik* 124(18):3756–3760 (Elsevier)
- Osuna-Enciso V, Cuevas E, Sossa H (2013) A comparison of nature inspired algorithms for multi threshold image segmentation. *Expert Syst Appl* 40:1213–1219
- Otsu N (1978) A threshold selection method from gray level histogram. *IEEE Trans Syst Man Cybern* 8(1):62–66
- Ouadfel S, Meshoul S (2013) A fully adaptive and hybrid method for image segmentation using multilevel thresholding. *Int J Image Graph Signal Process* 1:46–57
- Pei Z, Zhao Y, Liu Z (2009) Image segmentation based on differential evolution algorithm. In: International conference on image analysis and signal processing, Taizhou, IEEE Conference Proceedings, pp 48–51
- Peng B, Zhang L, Zhang D (2011) Automatic image segmentation by dynamic region merging. *IEEE Trans Image Process* 20(12):3592–3605
- Sathya PD, Kayalvizhi R (2012) Comparison of intelligent techniques for multilevel thresholding problem. *Int J Signal Imaging Syst Eng* 5(1):43–57
- Sharif M, Fathy M, Tayefeh Mahmoudi M (2002) A classified and comparative study of edge detection algorithms. In: International conference on information technology: coding and computing (ITCC), IEEE Conference Proceedings
- Sridevi M, Mala C, Sivasankar E, You I (2014) Optimized multilevel threshold selection using evolutionary computing. In: International conference on Network based information system, Italy, accepted for publication in IEEE conference proceedings
- Strang G, Nguyen T (1996) *Wavelets and filter banks*. Wellesley-Cambridge Press
- Yen JC, Chang FJ, Chang S (1995) A new criterion for automatic multilevel thresholding. *IEEE Trans Image Process* 4:370–378
- Yin PY, Chen LH (1997) A fast iterative scheme for multilevel thresholding methods. *Signal Process* 60:305–313
- Zahara E, Fan SKS, Tsai DM (2005) Optimal multi thresholding using a hybrid optimization approach. *Pattern Recognit Lett* 26:1082–1095
- Zhang GM, Chen SP, Liao JN (2011) Otsu image segmentation algorithm based on morphology and wavelet transformation. In: Proceedings of International conference on computer research and development (ICCRD), vol 1. IEEE, Shanghai, pp 279–283