DOI: 10.15849/IJASCA.221128.05

Int. J. Advance Soft Compu. Appl, Vol. 14, No. 3, November 2022
Print ISSN: 2710-1274, Online ISSN: 2074-8523

Multilevel Thresholding Image Segmentation Based-Logarithm Decreasing Inertia Weight Particle Swarm Optimization

Murinto, Adhi Prahara, Erik Iman Heri Ujianto

Informatic of Engineering, Universitas Ahmad Dahlan, Yogyakarta, 55166, Indonesia
murintokusno@tif.uad.ac.id

Informatic of Engineering, Universitas Ahmad Dahlan, Yogyakarta, 55166, Indonesia
adhi.prahara@tif.uad.ac.id

Informatic of Engineering, Universitas Teknologi, Yogyakarta, 55164, Indonesia
erik_iman@yahoo.com

Abstract

The image segmentatation technique that is often used is thresholding. Image segmentation is a process of dividing the image into different regions according to their similar characteristics. This research proposes a multilevel thresholding algorithm using modified particle swarm optimization to solve a segmentation problem. The threshold optimal values are determined by maximizing Otsu's objective function using optimization technique namely particle swarm optimization based on the logarithmic decreasing inertia weight (LogDIWPSO). The proposed method reduces the computational time to find the optimum thresholds of multilevel thresholding which evaluated on several grayscale images. A detailed comparison analysis with other multilevel thresholding based techniques namely particle swarm optimization (PSO), iterative particle swarm optimization (IPSO), and genetic algorithms (GA), From the experiments, Modified particle swarm optimization (MoPSO) produces better performance compared to the other methods in terms of fitness value, robustness and convergence. Therefore, it can be concluded that MoPSO is a good approach in finding the optimal threshold value.

Keywords**:** *grayscale image, inertia weight, image segmentation, particle swarm optimization.*

1 Introduction

Thresholding is a technique that can be done in an efficient and simple way for image segmentation in the aspect of implementation and processing time. Choosing a robust optimal threshold is an important step and challenging task in image segmentation.

Received 11 November 2021; Accepted 26 July 2022

Some reviews regarding the previous research of thresholding methods can be found in (Sahoo et al., 1988), (Dilpreet & Yadwinder, 2014) while the recent reviews of the methods can be found in (Guruprasad, 2020). The main purpose of image thresholding is to determine an optimal threshold value for the bi-level thresholding or several threshold values for the multilevel thresholding cases. The optimal threshold value is used to segment the pixels in an image into different groups. Along with the increasing problems in digital image processing field, multilevel thresholding becomes widely used to solve the problems. This mainly caused by its easy implementation and less memory consumption (Sathya & Kayalvizhi, 2010). Several studies of image segmentation using multilevel thresholding were carried out by (Rochmah et al., 2019; Dhieb & Frikha, 2016; Pare et al., 2017).

In recent years, heuristic optimization techniques are introduced in the field of image segmentation. This is due to the ability to complete computation quickly. Some of the meta-heuristic optimization techniques are differential evolution Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE) (Storn & Price, 1995), Ant Colony Optimization (ACO) (Dorigo et al., 2006), etc. PSO, first introduced by Kennedy and Eberhart (Eberhart & Kennedy, 1995), is a flexible, strong, and robust population-based stochastic search/optimization algorithm with inherent parallelism. This method has gained popularity compare with the other methods such as DE, ACO and GA because PSO provides superior search performance with faster and more stable convergence rates (P. Duraisamy & Kayalvizhi, 2010).

Although PSO is simple and easy to implement for image segmentation, the method may stuck/trapped in the local optimum just like the other optimization techniques. One of the solution to overcome this problem is by adding inertia weight to the PSO. The inertial weight plays a role in the process of providing a trade-off between diversification and intensification in the PSO algorithm. Particles will move while adjusting their speed and position according to the PSO equation, when inertial weights are used in the PSO algorithm. Shi and Eberhart were the first researchers to assign inertia weights to the PSO algorithm in 1998. The small inertia weight helps with the search space exploration while the large inertia weight facilitates search space exploitation. The proposed inertia weight uses random inertia weight initialization (Shi & Eberhart, 1998).

The use of inertia weight in PSO also has been carried out by several other studies. Linearly decreasing inertia weight strategy was introduced to improve the convergence speed of the PSO algorithm in the initial iteration of the search space (Xin et al., 2009). The inertia weight starts with some large values then decrease linearly to several smaller values. Inertia weight gives an excellent result from 0.9 to 0.4. Chen *et al.* (Chen et al., 2006) presented two natural exponent inertia weight strategies as e1-PSO and e2-PSO, which are based on the exponential decrease in inertia weight. Experimentally, this strategy is subjects to premature convergence, although the speed of convergence is fast towards the optimal position in the early stages of the search process. Malik *et al.* (Malik et al., 2007) presented sigmoid increasing inertia weight (SIIW) and sigmoid decreasing inertia weight (SDIW). These strategies provide better performance with fast convergence capability and narrowing aggressive movement towards the solution area. Oscillating inertia weight (Li & Gao, 2009) provides a balance between diversification and intensification waves and concludes that this strategy looks competitive and in some cases, performs better

in terms of consistency. Gao *et al.* (Gao et al., 2008) proposed exponential decreasing inertia weight (EDIW) with stochastic mutation (SM). Stochastic mutation (SM) was used to increase the diversity of swarm while EDIW was used to increase individual velocity convergence. Linearly decreasing inertia weight (LDIW) has been proposed by Shi and Eberhart (Shi & Eberhart, 1998) and greatly improves the accuracy and the speed of convergence. The large inertia weight facilitates the inclusive phase of the search space and then decreases linearly to a smaller inertia weight.

This paper introduces a new multilevel thresholding method based on inertia weight Particle Swarm Optimization (MoPSO) to solve multilevel thresholding problem in grayscale image segmentation. The validity of the proposed method was tested on 10 image samples and compared with the other methods namely PSO, GA and IPSO.

2 Multilevel Thresholding for Image Segmentation

The purpose of the optimization problem is to find a variable value that can optimize an objective function/fitness and at the same time satisfy the constraints. In this research, the fitness function is based on the Otsu function (Otsu, 1979). Otsu method is an inappropriate choice for multilevel image segmentation. Therefore, a new modified Otsu method based on Particle Swarm Optimization (MoPSO) is proposed to overcome the problem of multilevel thresholding. The main goal of the proposed method is to find the optimal thresholds for image segmentation by maximizing Otsu's objective function in smaller computation time.

Otsu method is a thresholding method that uses variance to measure the uniformity of the gray-level distribution of the image. Under certain conditions, this method is invariant to contrast and brightness of the image and it is taken as one of the effective methods among the automatic thresholding methods. This basic principle is used to segment the histogram into two groups based on certain threshold, one corresponds to the background while the other corresponds to the target. When the variance between the two groups is maximum, the optimal segmentation threshold is obtained.

In his research, Otsu (1979) defined the variance between classes as the sum of the Sigma function of each class, that can be computed using Equation (1).

$$g(t) = \sigma_0 + \sigma_1 \tag{1}$$

$$\sigma_0 = w_0 + (\mu_0 - \mu_A)^2$$

$$\sigma_1 = w_1 + (\mu_1 - \mu_A)^2$$

Where $\sigma_0 = w_0 + (\mu_0 - \mu_A)^2$ and $\sigma_1 = w_1 + (\mu_1 - \mu_A)^2$ are the variance of the first class and variance of the second class respectively. μ_T is the mean intensity of the original image. In a bilevel thresholding, the mean level of each class (μ_i) can be computed using Equation (2).

$$\mu_0 = \sum_{i=0}^{T-1} \frac{ip_i}{w_0}$$

$$\mu_1 = \sum_{i=T}^{L-1} \frac{ip_i}{w_1} \tag{2}$$

The optimal threshold is obtained from the maximization function between class variances, which can be computed using Equation (3).

$$T^* = argmax\ g(t) \tag{3}$$

While for the multithresholding problem, Otsu method can be written as in Equation (4).

$$\begin{aligned}\sigma_0 &= w_0 + (\mu_0 - \mu_A)^2 \\ \sigma_1 &= w_1 + (\mu_1 - \mu_A)^2 \\ \sigma_j &= w_j + (\mu_j - \mu_A)^2 \\ \sigma_m &= w_m + (\mu_m - \mu_A)^2 \end{aligned} \quad (4)$$

In multilevel thresholding, the mean level of each class (μ_i) can be computed using Equation below:

$$\mu_0 = \sum_{i=0}^{T_1-1} \frac{ip_i}{w_0}\ , \mu_1 = \sum_{i=T_1}^{T_2-1} \frac{ip_i}{w_1}, \mu_j = \sum_{i=T_j}^{T_{j+1}-1} \frac{ip_i}{\mu_j}, \mu_m = \sum_{i=T_n}^{L-1} \frac{ip_i}{\mu_n}$$

The optimal multilevel threhsold is obtained by maximizing the function of between class variances, which can be computed using Equation (5).

$$T^* = argmax\left(\sum_{i=0}^{n} \sigma_i\right) \quad (5)$$

3 Particle Swarm Optimization

The standard Particle Swarm Optimization (PSO) technique, first introduced by Kennedy and Eberhart in 1995 (Eberhart & Kennedy, 1995). It is a stochastic optimization technique similar to the behavior of birds flocking or the sociological behavior of a group of human. The basic idea of PSO is to involve a scenario where a flock of birds is in search of food sources within an area. All the birds do not know exactly where the food is, but with each iteration they will know how far the food will be found. The best strategy will be to follow the bird that is close to the food and also from the previous achieved best position. PSO is built with the concept of optimization through a particle swarm. Each particle lies at a position in the search space with a fitness value, evaluated through the fitness function to be optimized for each particle that represents the quality of that position. All particles will swarm through the multidimensional search space by adjsuting their position based on their own experience and their neighbors.

PSO consists of a group (swarm) of particles that are randomly initialized as points in the n-dimensional solution space in the search for an optimal solution to an optimization problem. In PSO, the population is also known as swarm, the candidate solutions are coded as particles in the search space. PSO starts with random intialization of the particles population. Particles move in the search space to find the optimal solution by updating the position of each particle based on their own experience and the surrounding particles. During movement, the current poisition of the i-th particle is represented by a vector $x_i = (x_{i1}, x_{i2}, .. x_{iD})$, where D is the search space dimension. The velocity of the i-th particle is represented as $v_i = (v_{i1}, v_{i2}, .. v_{iD})$. The best previous position of a particle is recorded as the personal best and named as pbest and the best global position obtained by the swarm is called gbest. PSO looks for the optimal solution by updating the position and velocity of each particle through the following Equation (Kennedy and Eberhart, 1995).

$$v_{id}^{t+1} = v_{id}^{t} + c_1 * r_{1i} * (p_{id} - x_{id}^{t}) + c_2 * r_{2i} * \left(p_{gd} - x_{id}^{t}\right) \quad (6)$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \quad (7)$$

Where t is the t-th iteration in the evolutionary process. $d \in D$ denotes the d-dimension in the search space. c_1 and c_2 are the acceleration constants that represent the weighting of the stochastic acceleration terms that push each particle to the $pbest\ (p_{id})$ and $gbest\ (p_{gd})$ positions. r_{1i} and r_{2i} are the random value that have a uniform distribution in the range of [0,1]. p_{id} and p_{gd} represent the elements of pbest and gbest in the d-dimension. v_{id} is limited by a predefined maximum velocity, $v_{max,d}$ to $[-v_{max}, v_{max}]$. $pbest_i = \left(pbest_i^1, pbest_i^2, \dots, pbest_i^d\right)$ represents the previous best position (the position that gives the best fitness value) of i-th particle. $gbest = (gbest^1, gbest^2, .. gbest^d)$ represents the best previous position of the population.

4 The Proposed Method

In this research, a modified particle swarm optimization algorithm (MoPSO) is proposed. The algorithm is based on the standard PSO. The difference is the initialization of particle positions uses chaotic variables to increase population diversification and ergodicity. In PSO, a larger inertia weight facilitates global exploration which allows the algorithm to find a new larger area, while a small inertia weight tends to facilitate local exploration. The inertia weight value that be used is obtained from the search of inertia weight using the logarithm decreasing inertia weight (LoDIW) (Gao et al., 2008). Empirical studies show that PSO with a large inertia weight value (w) have better global search capabilities compared to a smaller w with fast convergence. In his research, Gao (2008) introduced the logarithm decreasing inertia weight that can be computed using Equation (8).

$$w = \mathrm{w_{max}} + (\mathrm{w_{min}} - \mathrm{w_{max}})\ \mathrm{x}\ \log_{10}(\mathrm{a} + 10\mathrm{t}\ /\mathrm{T_{max}}) \quad (8)$$

Where a is a constant for the evoluationary velocity adjustment, here a = 1.

The update equation for velocity and chaotic position update in PSOt are written in Equation (9) and Equation (10) respectively.

Velocity update equation

$$v_{ij}^{t+1} = w * v_{ij}^t + c_1 . CF_1 * \left(P_{best,i}^t - x_{ij}^t\right) + c_2 . CF_2\left(G_{besti} - x_{ij}^t\right) \quad (9)$$

Position update equation

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^{t+1} \quad (10)$$

where $i = 1,2,3, \dots, n$ and $j = 1,2,3, \dots, d$

Where x_{ij}^t and v_{ij}^t denote position and velocity respectively. CF_1 and CF_2 are functions of the chaotic variable map, which replace the random values r_1 and r_2 in standard PSO.

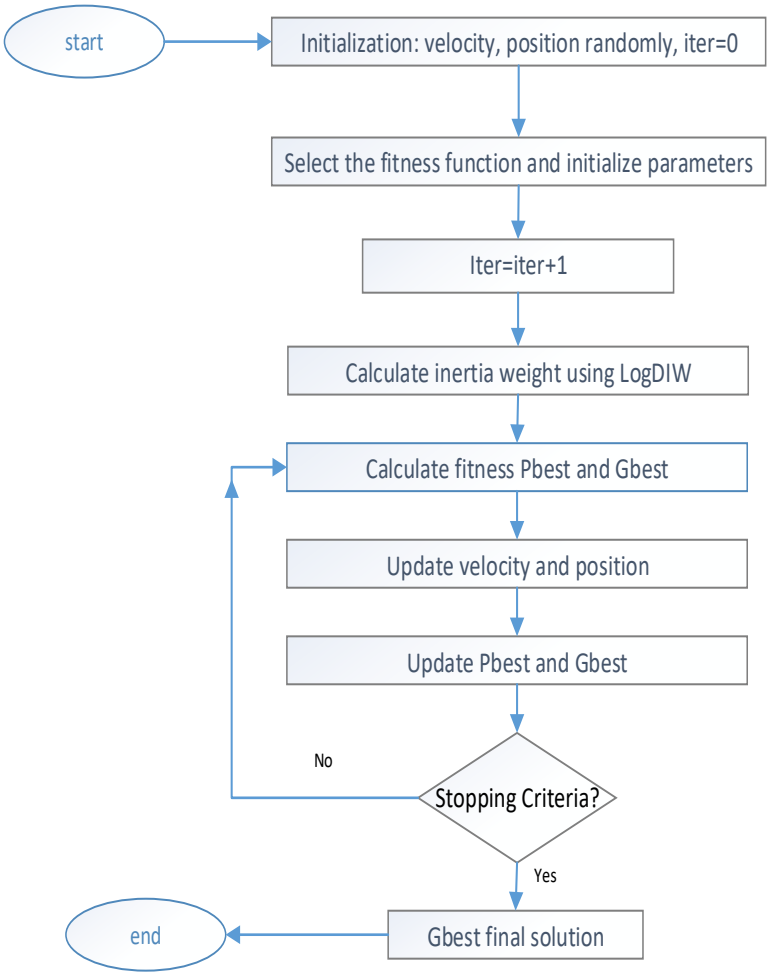

Figure 1. Flowchart of MoPSO based-inertia weight PSO

The pseudocode of the proposed modified Particle Swarm Optimization algorithm (MoPSO) can be explained in Algorithm 1.

Algorithm 1: The *pseudocode* of the proposed MoPSO

```
Begin
Initialize the position and velocity of the particles randomly.
Define the fit, as a fitness of the i-th particle.
While (the maximum number of iteration has not been reached) do
    For n = 1 to the number of particle do
        Determine the best previous position (pbest).
        Determine the best global position (gbest).
        Compute the inertia weight (w) using Equation (8).
        Update the position and velocity of the best global particle.
    End
    Generate next until the stopping criteria are met.
End
```

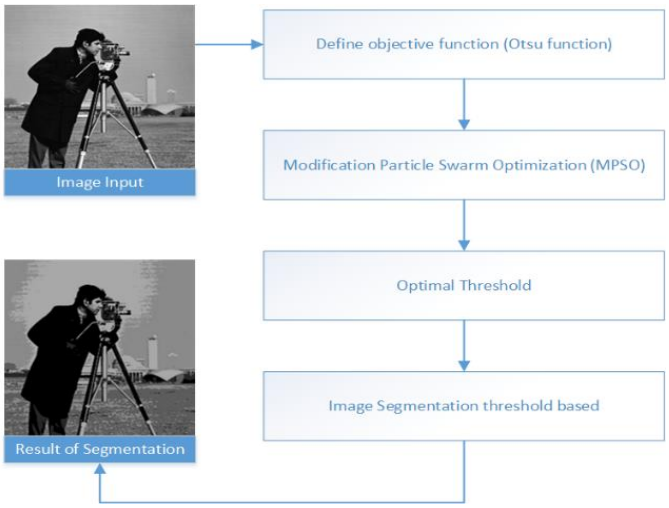

Figure 2. Diagram block of MoPSO bases image segmentation

5 The Experiment Result and Discussion

Experiments were conducted using various standard images with their histograms. The MoPSO algorithm which based on image threhsolding is implemented using Matlab programming language on a computer with specifications: Pentium ® Dual Core T4500 @2.30GHz and 2 GB of memory with Microsoft Windows 10 Operating System. Fitness value and CPU time are used in multilevel thresholding and compared with the other bio-inspired algorithms namely standard PSO (Kennedy J, 1995), Genetic Algorithm (GA) (Hammouche et al., 2008), and IPSO (Mo et al., 2011) respectively. The input image includes the Cameraman, Jetplane, Lake, Lena. Livingroom, Mandrill, Peppers, Pirate and their corresponding histograms are shown in Figure 3. The size of each image is 256 x 256 pixels.

The parameter of MoPSO algorithm for multilevel thresholding were determined through empirical analysis. The parameter values used in this research are shown in Table 1.

Table 1: Parameter values of MoPSO

Parameter	Value
Swarm size	40
Iteration	500
Run	20
w_{max}, w_{min}	0.9, 0.4
c_{max}, w_{min}	1

In this experiment, Otsu function is used as an objective function to evaluate the performance of the proposed algorithm. The objective value (fitness) of image segmentation that produced using multilevel thresholding based on Otsu function is shown in Table 2. The fitness function is used to determine whether the given threshold has reached the optimal value or not. The fitness value generated from image segmentation indicates that if the fitness value is higher, it will produce better segmentation. The result of image segmentation can be seen visually to see whether the segmentation quality are better or not in each threshold level that is used. Based on

Figure 4, it can be seen that between the quality level of m=2, 3, 4, 5, the segmentation quality looks better at the selected m = 5.

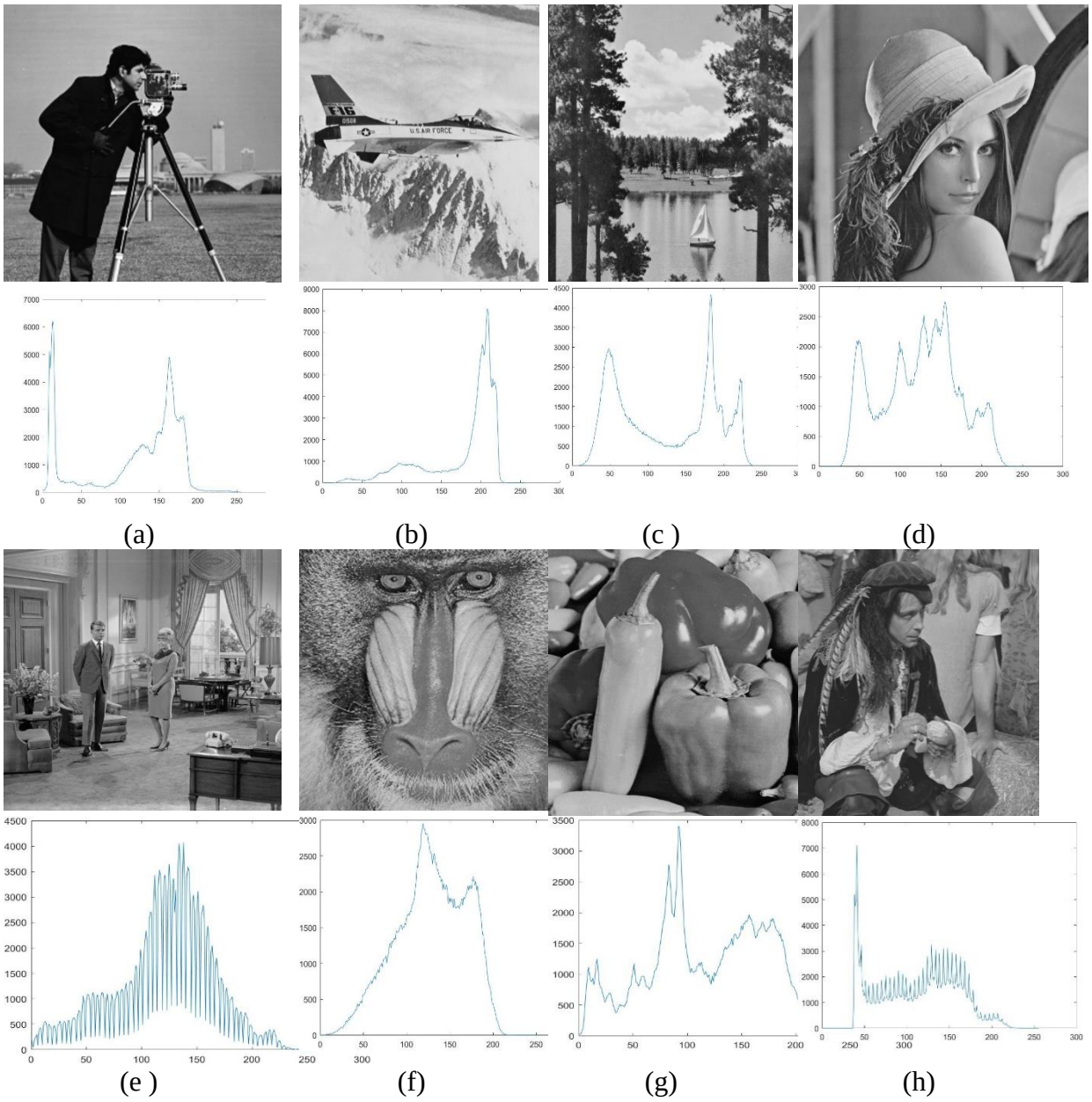

Fig 1. The test images and their corresponding histograms (a) Cameraman, (b) Jetplane, (c) Lake, (d) Lena, (e)

Table 2: Comparative analysis of the objective function obtained using Otsu method with the other multilevel thresholding methods.

Image	m	Fitness value				CPU Time			
		PSO	IPSO	MoPSO	GA	PSO	IPSO	MoPSO	GA
	2	3245,1514	3289,1159	3290,2330	3245,1514	0.7711	1.5903	0.69645	1.5942
	3	3609,5601	3650,3351	3651,8673	3609,5601	1.0134	1.6791	1.78994	1.7252
	4	3683,3513	3725,3265	3727,4142	3683,3513	1.1329	5.1118	1.64244	3.3356
Cameraman	5	3739,2401	3780,4443	3782,3976	3739,2670	1.3330	2.7090	2.30920	2.1898

	2	1770,2805	1770,2805	1770,2805	1770,2805	0.7697	0.7969	1.57607	1.7825
	3	1930,5758	1930,5758	1930,5758	1930,5758	0.9459	0.9819	5.53739	2.1172
	4	2007,2599	2007,2599	2007,2599	2007,2599	1.1307	1.1764	4.02599	4.3718
Jet plane	5	2053,9318	2053,7486	2051,0032	2053,9318	1.3090	1.3808	6.81593	3.3868
	2	3680,7370	3680,7370	3680,7370	3680,7370	0.7925	1.0854	1.84681	2.0980
	3	3970,3445	3970,3445	3970,3445	3970,3445	0.9604	1.4216	3.66987	5.3044
	4	4107,9656	4107,9656	4107,9656	4107,9656	1.1739	2.0902	4.88383	6.4020
Lake	5	4176,0226	4175,7568	4176,0754	4176,0754	1.3701	1.9329	3.89092	1.9621
	2	1601,2330	1770,2805	1601,2330	1601,2330	0.7892	2.6813	1.61633	3.1440
	3	1961,8014	1930,5758	1961,8014	1961,8014	0.9665	2.3401	3.54190	3.9938
	4	2128,2580	2007,2599	2128,2794	2128,2794	1.2487	4.8896	2.91227	5.8285
Lena	5	2191,7527	2050,0411	2191,8451	2191,8451	1.3157	1.9592	2.40609	8.1763
	2	1250,7151	1250,7151	1250,7151	1250,7151	0.7735	0.7944	1.52780	3.2698
	3	1627,9092	1627,9092	1627,9092	1627,9092	0.9521	1.1159	3.94691	3.8294
	4	1760,1030	1759,6741	1760,1030	1760,1030	1.1639	1.1751	7.01257	5.2266
Living room	5	1828,8644	1828,8644	1828,8644	1828,8644	1.3395	1.3679	6.09150	6.5257
	2	1069,1549	1069,1549	1069,1549	1069,1549	0.8474	0.8719	0.63284	3.8206
	3	1367,2782	1367,2782	1367,2782	1367,2782	1.2352	1.1026	2.13631	5.1686
	4	1454,5245	1454,4979	1454,5783	1454,5783	1.1292	2.8001	1.91632	4.6166
Mandrill	5	1503,9063	1503,9063	1503,9063	1503,9063	1.3236	2.8557	1.4225	8.9648
	2	2124,4103	2124,4103	2124,4103	2124,4103	0.7948	1.7919	1.4463	4.2863
	3	2510,8025	2510,8025	2510,8025	2510,8025	0.9588	2.2325	3.7037	6.9732
	4	2680,0965	2680,0965	2680,0965	2680,0965	1.1557	2.7953	4.6810	6.6846
Peppers	5	2737,9324	2736,9719	2737,9119	2737,9324	1.2875	3.2500	5.9839	8.0368
	2	1708,1202	1708,1202	1708,1202	1708,1202	0.7783	1.6901	0.7716	4.8780
	3	1994,5267	1994,5267	1994,5267	1994,5267	0.9506	2.1234	1.0139	7.9273
Pirate	4	2115,3044	2115,3056	2115,3044	2115,3056	1.1255	2.5439	1.9746	8.0889
	5	2169,6607	2169,7198	2163,3603	2169,7198	1.3211	3.0086	1.4692	6.6705

DOI: 10.15849/IJASCA.221128.05

Table 3: Comparative analysis of the threshold value obtained using Otsu based MoPSO with the other multilevel thresholding methods.

image	m	Threshold value			
		PSO	IPSO	MoPSO	GA
	2	87	87	88	**116**
	3	69 143 55 115	69 143	70 144	69 143
	4	152	56 115 153 40 93 138	57 116 154 41 94 139	55 115 153
Cameraman	5	0 0 68 89	168	169	0 0 56 88
	2	151	151	152	151
	3	111 171 88 141	111 171	112 172	111 171
	4	186 5 5 136	88 140 187 82 128 171	89 140 187 83 129 172	88 140 187
Jet plane	5	152	201	202	9 9 142 152
	2	124	125	125	124
	3	84 153 79 139	84 153	85 154	84 153
	4	193	77 139 193 66 109 157	8 140 194 67 110 158	77 139 192
Lake	5	0 0 90 122	197	198	3 3 83 127
	2	117	118	118	117
	3	92 150 79 125	92 150	93 151	92 150
	4	170	80 126 170 74 113 144	81 127 171 75 113 144	79 125 170
Lena	5	0 0 93 116	179	180	0 0 90 118
	2	105	106	106	105
	3	86 144 75 122	86 144	87 145	86 144
	4	160	75 122 162 55 96 131	76 123 163 56 97 132	75 122 161
Living room	5	0 0 85 104	167	168	0 0 78 112
	2	127	128	128	127
	3	97 147 84 122	97 147	98 148	97 147
	4	157 2 2 102	85 123 158 72 105 134	85 123 158 73 106 135	85 123 158
Mandrill	5	126	164	165	0 0 94 119
	2	117	117	117	116
	3	64 131 59 114	64 131	65 132	64 131
Peppers	4	161	60 115 162	61 116 163	60 114 161

Received 11 November 2021; Accepted 26 July 2022

			43 81 121	45 83 122	
	5	0 0 81 116	164	165	0 0 90 110
	2	107	108	108	107
	3	85 140	5 140	86 141	85 140
		69 112			
	4	153	70 113 154	71 114 155	69 112 154
Pirate			66 105 139	66 103 138	
	5	8 8 82 106	171	170	0 0 82 108

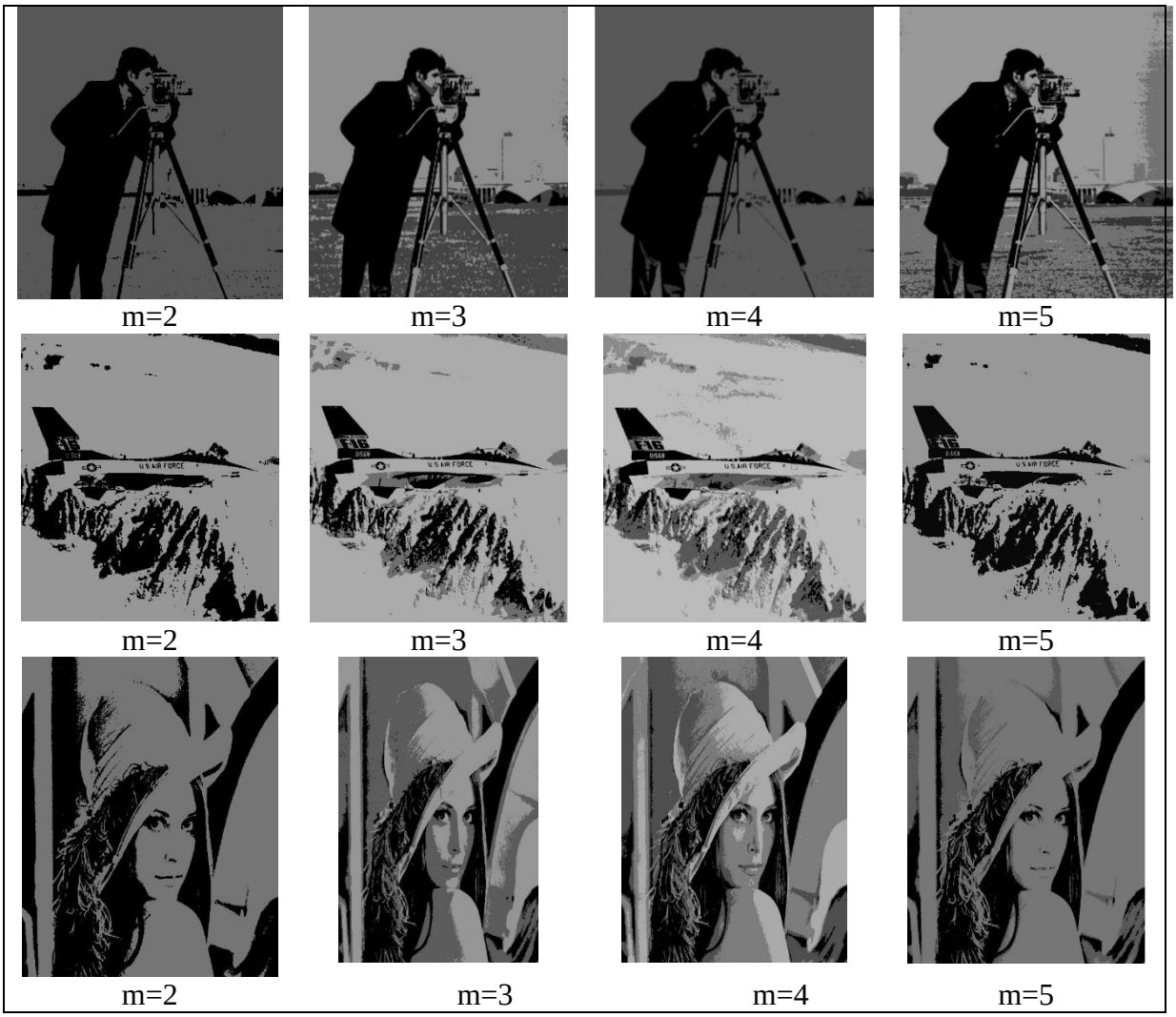

Fig 4. Grayscale image segmentation results using MoPSO at levels 2, 3, 4, 5 where (a) Cameraman, (b) Jetplane, (c) Lake, (d) Lena, (e) Livingroom, (f) Mandrill, (g) Peppers, and (h) Pirate.

The fitness value and computational time obtained from the Otsu function based on MoPSO are shown in Table 2. A high fitness value indicates that the segmentation result is good. On the other hand, if the fitness value is smaller, the segmentation result will be more unstable. Based on Table 2, by using MoPSO the result of image segmentation is more stable when compared to the other methods namely PSO, IPSO, and GA. Meanwhile, in terms of computational time, the MoPSO method is higher than the other methods.

6 Conclusions

In this research, a segmentation method for multilevel thresholding using modified particle swarm optimization (MoPSO) that based on the inertia weight has been proposed. The efficiency and effectivity of the proposed method are measured using eight standard images. The performance of the proposed method is compared with the other methods namely PSO, IPSO, and GA. From the experiments, MoPSO produces better performance compared to the other methods in terms of fitness value, convergence and robustness. Therefore, it can be concluded that MoPSO is a good approach in finding the optimal threshold value.

Acknowledgments

The authors are very grateful to Kemenristek Dikti for the financial support for this research in 2021.

References

[1]Chen, G., Huang, X., Jia, J., & Min, Z. (2006). *Natural Exponential Inertia Weight Strategy in Particle Swarm Optimization *. 2002*, 3672–3675.

[2]Dhieb, M., & Frikha, M. (2016). A multilevel thresholding algorithm for image segmentation based on particle swarm optimization. *Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA*, *0*. https://doi.org/10.1109/AICCSA.2016.7945752

[3]Dilpreet, K., & Yadwinder, K. (2014). Various Image Segmentation Techniques: A Review. *International Journal of Computer Science and Mobile Computing*, *3*(5), 809–814.

[4]Dorigo, M., Stutzle, T., & Birattari, M. (2006). Ant Colony Optimization. *IEEE Computational Intelligence Magazine*, *November 2006*, 28–39. https://doi.org/10.1007/978-3-030-67380-2_2

[5]Eberhart, R., & Kennedy, J. (1995). New optimizer using particle swarm theory. *Proceedings of the International Symposium on Micro Machine and Human Science*, 39–43. https://doi.org/10.1109/mhs.1995.494215

[6]Gao, Y. L., An, X. H., & Liu, J. M. (2008). A particle swarm optimization algorithm with logarithm decreasing inertia weight and chaos mutation. *Proceedings - 2008 International Conference on Computational Intelligence*

and Security, CIS 2008, *1*, 61–65. https://doi.org/10.1109/CIS.2008.183

[7]Guruprasad, P. (2020). Overview Of Different Thresholding Methods In Image Processing. *TEQIP Sponsored 3rd National Conference on ETACC*, *June*.

[8]Hammouche, K., Diaf, M., & Siarry, P. (2008). A multilevel automatic thresholding method based on a genetic algorithm for a fast image segmentation. *Computer Vision and Image Understanding*, *109*(2), 163–175. https://doi.org/10.1016/j.cviu.2007.09.001

[9]Kennedy J, E. R. (1995). Particle swarm optimization. *IEEE International Conference on Neural Networks*, 1942–8.

[10] Li, H. R., & Gao, Y. L. (2009). Particle swarm optimization algorithm with exponent decreasing inertia weight and stochastic mutation. *2009 2nd International Conference on Information and Computing Science, ICIC 2009*, *1*(3), 66–69. https://doi.org/10.1109/ICIC.2009.24

[11] Malik, R. F., Rahman, T. a, Hashim, S. Z. M., & Ngah, R. (2007). New Particle Swarm Optimizer with Sigmoid Increasing Inertia Weight. *International Journal of Computer Science and Security. IJCSS*, *1*(1), 35–44.

[12] Mo, S., Zeng, J., & Tan, Y. (2011). Particle swarm optimisation based on self-organisation topology driven by different fitness rank. *International Journal of Computational Science and Engineering*, *6*(1–2), 24–33. https://doi.org/10.1504/IJCSE.2011.041209

[13] Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. *IEEE TRANSACTIONS ON SYSTREMS, MAN, AND CYBERNETICS*, *SMC-9*(1), 62–66.

[14] Duraisamy, S., & Kayalvizhi, R. (2010). A New Multilevel Thresholding Method Using Swarm Intelligence Algorithm for Image Segmentation. *Journal of Intelligent Learning Systems and Applications*, *02*(03), 126–138. https://doi.org/10.4236/jilsa.2010.23016

[15] Pare, S., Bhandari, A. K., Kumar, A., & Singh, G. K. (2017). An optimal color image multilevel thresholding technique using grey-level co-occurrence matrix. *Expert Systems with Applications*, *87*, 335–362. https://doi.org/10.1016/j.eswa.2017.06.021

[16] Murinto., Rochmah, N., Puji, D., & Miksa, M. (2019). *Multilevel thresholding hyperspectral image segmentation based on independent component analysis and swarm optimization methods*. *5*(1), 66–75.

[17] Sahoo, P. K., Soltani, S., & Wong, A. K. C. (1988). A survey of thresholding techniques. *Computer Vision, Graphics and Image Processing*, *41*(2), 233–260.

[18] Sathya, P. D., & Kayalvizhi, R. (2010). Development of a New Optimal Multilevel Thresholding Using Improved Particle Swarm Optimization Algorithm for. *International Journal of Electronics*, *2*(1), 63–67.

[19] Shi, Y., & Eberhart, R. (1998). Modified particle swarm optimizer. *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC*, *February*, 69–73.

[20] Storn, R., & Price, K. (1995). Differential evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces. *Technical Report*, *TR-95-012*, 1–12.

[21] Xin, J., Chen, G., & Hai, Y. (2009). A particle swarm optimizer with multi-stage linearly-decreasing inertia weight. *Proceedings of the 2009 International Joint Conference on Computational Sciences and Optimization, CSO 2009*, *1*, 505–508.

Authors Profile

Murinto, received his M.Kom from Computer Sciences Gadjah Mada University, Yogyakarta in 2004 and Doctor (Dr) from Computer Sciences Gadjah Mada University in 2021. Currently working as Assistant Professor in Universitas Ahmad Dahlan in Yogyakarta also. He is senior researcher in the college, his topics research is image processing, computer vision, machine learning.

Adhi Prahara, received his S.Si and M.Kom from Department Computer Sciences and Electronics Gadjah Mada University, Yogyakarta. Currently working as lecturer in Universitas Ahmad Dahlan in Yogyakarta also. He is researcher in the college, his topics research is image processing, computer vision, machine learning

Erik Iman Heri Ujianto, received his M.Kom from Computer Sciences and Doctor from Computer Sciences Gadjah Mada University, Yogyakarta. Currently working as Associate Professor in Universitas Teknologi Yogyakarta also. He is researcher in the college, his topics research is image processing.