IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received December 20, 2018, accepted January 1, 2019, date of publication January 10, 2019, date of current version February 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2891673

Modified Grasshopper Algorithm-Based Multilevel Thresholding for Color Image Segmentation

HONGNAN LIANG, HEMING JIA, (Member, IEEE), ZHIKAI XING, JUN MA, AND XIAOXU PENG
College of Mechanical and Electrical Engineering, Northeast Forestry University, Harbin 150040, China

Corresponding author: Heming Jia (jiaheming@nefu.edu.cn)

ABSTRACT Multilevel thresholding is an important approach for image segmentation which has drawn much attention during the past few years. The Tsallis entropy method is implemented for its effectiveness and simplicity. Although it is efficient and gives an excellent result in the case of bi-level thresholding, its evaluation becomes complexity when the number of thresholds increases. To overcome the problem, the metaheuristic algorithms are applied in this search area for searching the optimal thresholds. In this paper, a modified grasshopper optimization algorithm (GOA) is adopted to render multilevel Tsallis cross entropy more practical and reduce the complexity. The Levy flight algorithm is employed to modify the original GOA and balance the exploration and exploitation of the GOA. Experiments are conducted between five state-of-the-art metaheuristic algorithms and the proposed one. In addition, the proposed approach is compared with thresholding techniques depending on between-class variance (Otsu) method and the Renyi entropy function. Both real life images and plant stomata images are used in the experiments to test the performance of the algorithms involved. Qualitative experimental results show that the proposed segmentation approach has a fewer iterations and a higher segmentation accuracy.

INDEX TERMS Multi-threshold color image segmentation, Tsallis entropy method, grasshopper optimization algorithm, Levy flight.

I. INTRODUCTION

Image segmentation is a fundamental field in image analysis science, which plays an important role in image processing. There are primarily four types of segmentation methods: thresholding, boundary-based, region-based, and hybrid techniques [1]–[5]. Boundary-based methods assume that the pixel properties, such as intensity, color, and texture, change abruptly between different regions [6]–[10]. Region-based methods assume that neighboring pixels within the same region should have similar values [11]–[13]. Hybrid methods tend to combine boundary detection and region growing together to achieve better segmentation.

Among image segmentation methods, thresholding segmentation has received widely attention because of its simplicity, small storage space, fast processing speed, and ease in manipulation [14]. Thresholding methods involve selecting a set of thresholds using some characteristics defined from images [15]–[17]. Multilevel thresholding is an extension of bi-level thresholding for separating the color image into more than two classes. Among the thresholding techniques, Tsallis [18] and Otsu methods are the most popular ones. Otsu method maximizes the between class variance function, whereas, Kapur method maximizes posterior entropy of the segmented classes to find optimum thresholds [19]. Computational complexity of Tsallis and Otsu methods increases exponentially with the increasing numbers of thresholds due to exhaustive search.

Due to the small difference between the target and the background of a complex image, the bi-level thresholding cannot exactly find the optimal threshold. The multi-threshold image can be divided into multiple regions to find the foreground and background in the image. Especially, under the circumstance of segmenting complex images, such as medical image, satellite image and plant image segmentation, the multi-threshold image segmentation method can segment the target region accurately. However, as the number of threshold increases, the computed amount of algorithm increases and the operation time becomes slow. It is a critical and challenging task for traditional exhaustive methods because of the high computational costs. In such case,

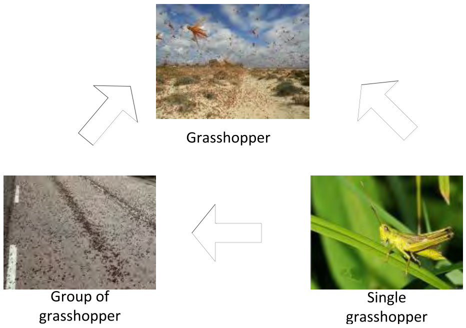

FIGURE 1. Grasshopper growth cycle.

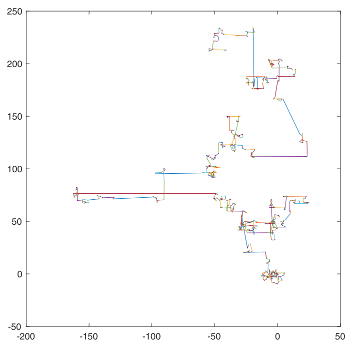

FIGURE 2. Levy's flight path.

TABLE 1. Parameters and references of the comparison algorithms.

Algorithm	Parameters	Value
GOA	c_{max}	1
	c_{min}	0.0004
WOA[56]	a	[0.2]
	b	1
	l	[-1,1]
FPA[57]	P	0.5
PSO[58]	Swam size	200
	Cognitive, social acceleration	2,2
	Inertial weight	0.95-0.4
BA[59]	β	(0,1)
MGOA	Levy	0.8

FIGURE 3. The test images.

metaheuristic methods drew much attention in recent years. In 2018, Raja *et al.* [20] proposed firefly algorithm and Tsallis entropy based approach. The proposed method offers an average enhancement of cluster classification by 4.44% in terms of silhouette index. In 2016, Alva *et al.* [21] proposed Half-life Constant Particle Swarm Optimization and Tsallis entropy based approach. This method solves multidimensional problems. In 2016, Sarkar *et al.* [22] proposed a novel approach for unsupervised classification of land cover study of hyper-spectral satellite images to improve separation between objects and background by using multilevel thresholding based on the maximum Renyi entropy. He and Huang [23] proposed a modified firefly algorithm (MFA) to find the optimal multilevel threshold values for a color image. The MFA algorithm is an effective method for multilevel color image thresholding segmentation. In 2016, Muangkote *et al.* [24] proposed an improved version of the moth-flame optimization (MFO) algorithm for image segmentation to effectively enhance the optimal multilevel thresholding of satellite images. Experimental results indicate that the MTMFO more effectively and accurately identifies the optimal threshold values. In 2018, Pare *et al.* [25] proposed a modified fuzzy entropy (MFE) functional. MFE function is the difference of adjacent entropies, which is optimized to provide thresholding levels, such that all regions have almost equal entropies. V. K. Bohat [26] proposed a novel thresholding (TH) heuristic for multilevel thresholding problem. The proposed algorithm has higher image segmentation accuracy and shorter CPU time.

Therefore, traditional entropy based criterion has been coupled with different meta-heuristic techniques to improve the performance of multilevel thresholding in terms of stability and threshold selection [27]–[29]. Metaheuristics algorithms have handle the optimization problems by mimicking physical or biological phenomena. In the last couple of years, few works have been accomplished in favor of multilevel segmentation of colored images due to the exponentially increasing complexities involved in the computation of threshold values. Evolutionary techniques such as an ant colony algorithm (ACO) [30]. ACO imitated ants to find the shortest path social behavior, Particle Swarm Optimization (PSO) algorithm [31]–[34] simulated the behavior of birds

TABLE 2. The optimal fitness value and threshold value of each algorithm under Tsallis.

K	GOA				WOA				FPA			
	R	G	B	F	R	G	B	F	R	G	B	F
Test1												
4	40 80 161	46 89 137	48 86 132		40 90 162	46 89 137	48 87 133		38 89	45 89	48 81	
	227	208	206	26.899	227	208	207	26.2306	153 223	138 208	131 208	25.8427
8	18 39 62	27 50 72	29 49 69		18 42 69	28 52 74	27 47 68		28 50	29 56	32 54	
	85 111 146	95 120	90 113		97 131 178	97 121	90 115		97 103	83 92	78 94	
	191 232	147 182	141 179		218 243	148 183	143 181		126 171	118 141	121 158	
		228	227	30.9266		229	228	30.1564	229 232	179 226	184 221	29.2066
10	14 32 51	23 42 60	22 39 55		5 21 41 62	22 41 60	26 43 59		15 45	27 47	26 48	
	72 94 119	79 98 118	72 90 111		87 118 157	77 96 115	75 92 111		61 83	48 65	65 88	
	151 188	139 164	133 162		193 217	138 163	133 161		92 93	82 96	101 105	
	219 242	196 233	196 233		241	196 235	196 235		120 145	127 170	147 169	
				34.3662				34.3542	188 233	193 226	221 225	33.5411
12	12 27 44	19 33 47	21 34 47		14 30 47	19 36 52	20 33 47		19 21	22 42	28 33	
	62 81 100	59 74 90	61 74 88		65 83 100	67 83 99	62 77 92		41 59	63 71	35 49	
	120 139	107 124	103 118		124 152	116 134	109 128		92 110	82 104	57 79	
	164 190	144 169	134 157		182 208	153 177	149 175		136 152	121 151	95 102	
	217 240	201 237	188 231		228 246	208 239	207 241		196 203	191 199	117 151	
				37.9479				37.4725	230 238	218 232	166 230	35.9612
Test2												
4	115 161	103 140	62 93 119		113 161	106 141	60 91 118		110 161	100 139	63 93	
	181 210	161 185	144	25.5954	181 210	162 186	144	25.5599	181 210	161 185	119 144	25.3401
8	74 116 145	64 105	28 61 81		70 106 109	16 51 103	20 57 80		69 109	43 80	29 66	
	161 173	128 143	97 114		151 167	129 145	96 112		134 153	106 128	85 101	
	185 202	156 168	133 148		181 197	159 171	128 142		167 180	144 156	120 137	
	225	184 208	177	29.6095	221	191	153	29.5949	196 220	169 188	150 256	28.334
10	38 73 92	57 80 99	42 54 57		81 120 134	66 80 111	10 19 39		54 86	24 37	1 32 58	
	99 138 158	119 135	110 111		154 164	131 144	52 78 94		105 124	59 94	74 86	
	171 183	147 158	133 185		174 184	154 163	111 128		148 162	121 140	99 115	
	199 223	170 186	216 233		195 212	172 186	141 152		173 184	154 167	130 146	
		221	247	33.2135	232	209		33.0286	199 220	184 210	175	32.9939
12	55 62 86	26 45 75	60 62 111		34 42 59	52 80 112	4 20 31 39		73 98	59 83	21 35	
	99 128 147	107 128	114 142		112 137	128 140	64 81 94		109 131	97 114	55 72	
	161 171	143 156	167 177		153 165	149 157	107 120		146 157	129 140	85 98	
	182 194	168 182	183 184		174 184	165 173	133 144		166 175	150 160	113 127	
	211 231	197 231	229 250	36.5818	194 210	185 199	154	36.4021	184 195	170 183	138 148	35.2611
		242	254		233	221			212 233	202 236	156 256	
Test3												
4	40 70 105	43 76 108	72 119		40 71 106	43 77 109	74 121		40 71	43 76	73 120	
	151	147	155 187	27.3472	152	148	156 187	27.3268	106 152	108 147	155 187	26.8225
8	26 46 63	32 56 75	46 73 103		27 42 55	23 42 61	46 73 105		26 45	27 51	44 69	
	81 102 123	92 112	131 152		71 90 111	77 94 114	132 153		61 80	71 90	98 128	
	148 178	135 163	169 188		136 170	136 164	171 189		102 129	111 134	151 169	
		256	211	31.4121			213	34.4113	167 256	162 256	188 212	30.4256
10	23 39 52	21 40 58	13 42 62		17 30 45	8 13 29 42	3 44 72 99		28 47	1 4 21	32 52	
	67 84 103	72 85 99	86 112		58 72 89	61 77 93	123 143		63 80	40 58	75 102	
	122 143	115 133	135 154		108 129	113 135	160 176		98 117	75 92	127 147	
	165 183	152 174	171 190		155 187	164	193 215		140 171	112 133	164 184	
			213	35.1473				35.1466	236 256	162	207 244	34.6877
12	16 31 44	17 33 49	19 37 51		21 35 47	17 20 29	14 28 49		3 14 29	16 29	1 16 27	
	56 69 84	62 76 90	66 86 107		58 70 82	45 59 73	70 97 120		44 56	43 63	42 56	
	99 118 136	106 123	127 145		95 109 124	85 98 115	139 154		70 85	90 115	69 81	
	159 180	140 160	160 176		142 163	132 151	167 180		102 120	134 150	95 112	
	197	177 199	193 216		189	175	197 217		138 166	164 179	132 153	
				38.6498				38.6486	240	195 215	179	37.6298
Test4												
4	103 136	93 126	86 120		103 136	93 126	74 105		111 150	92 125	74 105	
	167 195	157 185	151 231	26.4604	167 195	157 185	131 157	26.4573	185 221	155 184	131 157	26.3345
8	51 84 108	66 98 130	48 71 89		76 99 118	67 89 110	30 63 84		75 100	1 1 85	1 46 74	
	129 149	159 186	106 121		138 157	128 147	102 121		120 140	114 142	96 115	
	170 189	244 253	136 152		176 192	165 181	136 153		159 176	167 191	132 148	
	208	253	171	30.5352	210	200	172	30.4207	192 209	248	167	30.1312
10	44 45 71	65 85 102	39 62 78		51 69 96	49 75 93	48 57 73		27 72	1 65 88	1 57 81	
	89 108 128	118 135	93 107		116 134	110 128	87 102		96 115	107 128	102 121	
	149 170	152 167	122 135		151 167	146 161	117 131		134 153	149 168	138 160	
	189 208	182 199	149 166		182 196	175 190	143 158		172 190	187 206	224 256	
		249	190	34.1714	213	204	176	34.4586	208 236	243	256	34.0771
12	33 42 57	1 6 33 59	26 47 64		56 84 100	40 51 77	1 35 49 69		36 67	1 64 86	1 1 42	
	80 99 117	81 100	79 93 105		115 130	94 109	84 99 112		92 111	104 122	63 80	
	134 151	119 136	117 128		143 156	125 140	124 136		129 147	139 155	95 109	
	167 182	153 168	138 149		169 181	154 167	147 161		164 179	170 184	121 134	
	195 211	183 200	162 176		192 204	179 192	177		193 209	200 245	146 160	
				37.8453	217	205		37.8041	248 255	255	176	37.5706
Test5												
4	51 91 147	50 82 135	51 81 132		50 90 147	51 83 136	51 81 132		51 91	51 83	51 81	
	213	206	205	26.5968	213	206	205	26.579	148 214	135 205	132 205	26.3367
8	34 49 69	42 55 72	41 54 67		35 50 70	42 56 72	42 54 67		32 45	42 55	38 50	
	96 122 154	93 119	85 110		93 119 150	93 119	85 110		62 86	71 92	63 82	
	188 226	150 186	142 180		187 230	150 186	143 181		115 152	118 149	113 150	
		227	225	30.7281		229	226	30.7126	192 232	185 226	186 220	28.8702
10	32 42 55	39 49 61	9 40 52		31 45 59	39 51 62	12 41 52		32 43	40 52	40 52	
	72 95 118	77 97 120	65 82 107		80 108 131	77 98 117	63 77 96		56 76	66 85	64 82	
	144 172	144 166	138 175		145 173	142 171	120 150		100 126	110 137	109 145	
	200 233	191 226	205 231		203 236	201 236	185 228		155 184	164 187	187 218	
				34.4467				34.2253	205 233	208 233	233 244	32.5953
12	30 39 48	38 47 56	36 44 53		30 39 49	38 48 58	37 46 55		26 32	36 45	36 46	
	61 77 94	67 83 102	62 73 89		62 78 97	70 83 99	64 76 91		40 49	54 65	56 66	
	111 126	125 151	110 137		117 137	117 138	107 127		62 81	81 103	80 99	
	140 162	173 194	167 197		160 184	161 182	151 177		102 124	133 166	119 141	
	191 228	214 233	223 237		208 237	209 239	206 236		152 180	196 212	161 178	
				37.9681				37.9035	211 239	232 243	201 230	36.4008
Test6												
4	54 82 119	57 83 108	32 56 80		59 106 133	48 101	52 104		54 82	57 82	34 58	
	191	175	141	26.7791	153	149 177	137 158	26.7779	118 188	107 176	82 144	26.7459
8	39 61 81	40 58 76	24 42 60		26 57 86	40 73 98	23 47 81		1 9 38	45 65	19 34	
	106 137	91 108	78 104		106 122	128 148	108 126		59 76	83 99	49 65	
	180 201	138 176	134 167		136 149	164 178	141 153		95 126	117 158	84 126	
	225	218	191	30.7739	162	193	166	30.7653	188	197 229	181 256	29.9481
10	30 46 61	36 52 65	19 35 50		26 50 65	18 44 76	9 21 34 61		30 45	33 48	17 32	
	76 93 116	78 91 103	66 83 109		90 108 122	108 132	95 119		58 71	64 78	47 64	
	142 164	120 149	137 148		134 145	149 163	132 144		84 100	92 107	82 118	
	180 217	170 215	162 179		155 165	176 187	156 168		126 159	135 181	160 192	
				34.5693		198		34.5847	206 240	225 237	201 256	34.2141
12	25 39 53	39 56 74	13 25 36		1 10 28 47	37 43 71	7 23 39 64		19 31	37 52	5 17 29	
	66 78 91	89 106	47 59 70		67 91 109	102 126	82 100		43 55	66 79	40 53	
	109 132	136 175	82 99 114		123 134	140 151	115 125		67 79	91 105	67 84	
	165 199	182 187	141 167		145 155	164 176	135 146		94 113	122 151	137 157	
	203 224	192 201	193		166	184 189	157 169		135 173	173 194	195 256	
		217		38.0844		198		38.3809	209 239	210 240	256	37.0039

TABLE 2. ***(Continued.)*** **The optimal fitness value and threshold value of each algorithm under Tsallis.**

Leaf1												
4	79 95 114	95 112	52 72 160		79 95 114	96 113	11 52 72		78 94	95 112	48 63	
	190	131 199	246	25.5968	192	132 199	168	24.1273	113 188	131 199	81 164	23.3367
8	68 78 87	91 105	42 53 63		67 78 87	84 96 106	39 48 57		67 78	85 96	42 53	
	95 104 117	118 135	76 95 228		96 107 121	116 127	65 75 89		87 96	105 114	64 77	
	140 221	181 184	229 240		151 199	146 172	126 192		106 118	124 137	97 171	
		227 229		30.1141		208		27.1507	144 191	161 202	179 230	24.8702
10	71 82 92	83 95 105	39 48 57		41 72 84	83 94 103	14 40 50		67 78	84 95	44 57	
	104 120	115 127	66 78 94		95 108 127	112 122	59 68 80		87 97	105 116	70 88	
	178 179	145 182	135 191		200 235	134 152	99 164		109 130	130 153	169 219	
	183 184	185 224	237 240		254 254	167 167	215 247		172 185	164 174	237 239	
	231	228		34.2417		220		31.2286	202 211	176 208	241 245	30.5213
12	64 71 78	5 13 24	39 49 57		41 65 73	80 89 97	22 37 45		69 81	77 86	11 41	
	85 92 99	27 29 84	66 76 93		81 87 93	105 114	52 59 66		91 102	94 101	52 63	
	107 118	103 124	130 135		101 111	122 133	73 82 94		118 158	108 115	76 95	
	130 149	149 159	135 151		121 135	145 154	121 195		180 208	123 134	139 154	
	191 220	180 199	164 240		164 218	171 219	249		210 212	149 171	176 222	
				37.1281		219		35.2141	220 221	217 223	235 240	34.4118
Leaf2												
4	108 120	131 143	74 86 97		108 120	132 145	74 86 98		107 120	132 144	74 86	
	132 146	154 168	112	26.0191	132 146	156 171	112	24.1245	132 146	156 170	98 112	25.7024
8	99 107 115	10 60 95	100 105		100 108	124 133	66 74 81		100 109	29 29	17 41	
	122 129	108 121	129 134		116 123	140 146	88 95 102		117 124	30 52	88 117	
	137 146	175 177	152 198		130 137	152 159	111 124		131 140	59 93	122 129	
	159	236	207 249	30.2439	146 159	168 180		27.2216	152 177	179 197	159 163	28.4571
10	25 26 35	15 30 38	39 86 88		61 77 79	110 124	65 72 78		25 68	9 62 71	33 39	
	43 92 119	65 88 89	105 132		86 93 98	132 139	84 90 96		82 82	115 123	44 56	
	151 177	95 139	134 154		108 113	146 152	102 109		88 131	128 157	72 124	
	218 252	204 222	192 202		117 152	158 165	118 132		158 177	161 164	161 172	
			246	34.3193		173 185		31.0243	193 201	178	219 255	32.2211
12	5 28 40 71	21 34 64	15 79 102		98 106 113	96 103	122 130		14 14	21 46	4 15 49	
	76 125 128	68 88 108	102 119		119 124	109 115	137 143		39 94	113 125	51 145	
	144 144	113 164	123 131		129 136	120 125	148 152		119 145	145 148	161 163	
	149 169	202 202	142 173		143 154	130 135	152 157		169 171	160 174	171 234	
	201	244 252	187 224		166 166	141 148	163 169		181 214	184 186	238 243	
			238	38.1144	166	157 171	179 190	35.4215	214 254	211 223	249	35.0039

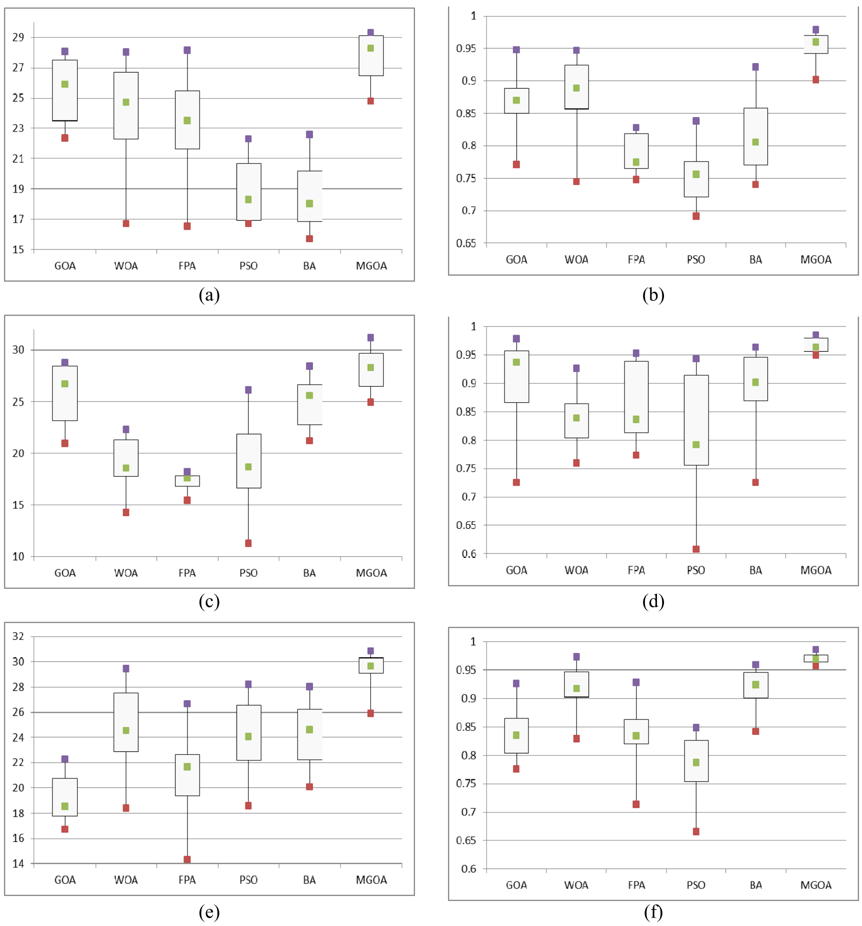

FIGURE 4. **Box-plot representing the PSNR and FSIM for all methods. (a) PSNR under Tsallis. (b) FSIM under Tsallis. (c) PSNR under Otsu. (d) FSIM under Otsu. (e) PSNR under Renyi. (f) FSIM under Renyi.**

in navigation and hunting. Other group optimization algorithms are: Artificial Bee Colony Algorithm proposed by Karaboga in 2015 [35], [36], which could be optimized by imitating the behavior of bees to collect nectar; The Flower Pollination Algorithm (FPA) [37] is proposed by Xin-she Yang. This algorithm is inspired by the pollination process

TABLE 3. The optimal fitness value and threshold value of each algorithm under Tsallis.

K	PSO				BA				MGOA			
	R	G	B	F	R	G	B	F	R	G	B	F
Test1												
4	32 62 105 189	32 62 105 189	57 101 111 202	24.8199	39 88 160 226	43 84 132 205	46 84 131 205	23.8157	40 80 161 227	46 89 137 208	48 86 132 206	26.8991
8	24 54 86 117 174 215 231 236	29 56 91 109 117 143 182 217	31 43 50 70 104 123 154 220	29.1546	17 38 60 84 110 143 188 232	28 51 73 95 120 147 182 228	3 10 61 193 196 211 215 232	26.2186	18 39 62 85 111 146 191 232	27 50 72 95 120 147 182 228	29 49 69 90 113 141 179 227	30.9117
10	22 52 71 89 99 138 175 208 240 240	25 41 46 66 76 116 136 161 183 226	28 45 60 62 63 74 104 121 157 222	32.3442	8 23 41 61 84 110 143 182 216 239	2 6 107 122 131 160 166 205 237 240	21 36 53 70 88 107 130 157 192 233	31.5851	14 32 51 72 94 119 151 188 219 242	23 42 60 79 98 118 139 164 196 233	22 39 55 72 90 111 133 162 196 233	34.4059
12	21 51 70 73 85 107 133 153 163 171 218 242	23 49 67 90 106 117 118 144 146 185 215 228	25 44 65 68 84 105 107 114 128 161 180 225	34.9119	7 18 31 45 60 78 98 122 154 189 219 242	16 27 40 54 68 84 101 118 137 162 196 233	15 27 39 53 68 84 101 119 140 164 195 231	32.9552	12 27 44 62 81 100 120 139 164 190 217 240	19 33 47 59 74 90 107 124 144 169 201 237	21 34 47 61 74 88 103 118 134 157 188 231	**38.1014**
Test2												
4	124 159 179 211	84 138 164 184	56 96 122 143	23.0554	32 50 100 129	99 138 160 184	62 93 119 144	22.3771	115 161 181 210	103 140 161 185	62 93 119 144	25.5935
8	116 129 160 170 184 185 204 226	82 119 125 140 157 162 168 191	46 57 83 88 97 109 129 149	26.6115	98 141 157 169 180 192 209 230	35 41 69 69 70 76 179 254	24 51 74 89 103 120 136 149	25.3991	74 116 145 161 173 185 202 225	64 105 128 143 156 168 184 208	28 61 81 97 114 133 148 177	29.6095
10	103 138 138 155 168 171 181 199 208 228	46 114 128 144 151 160 170 174 182 195	41 73 82 90 91 102 120 137 146 150	31.2325	76 113 138 153 164 173 183 195 211 230	15 65 97 118 134 146 158 169 185 213	21 40 61 75 87 98 112 127 140 151	30.1139	38 73 92 99 138 158 171 183 199 223	57 80 99 119 135 147 158 170 186 221	42 54 57 110 111 133 185 216 233 247	33.2294
12	95 98 106 134 151 168 174 183 191 211 219 231	77 97 108 131 142 147 158 168 185 193 206 215	33 64 84 101 115 131 136 139 146 146 149 154	36.5818	16 76 86 88 122 123 125 158 181 207 223 230	53 64 85 107 125 138 149 159 169 184 206 232	2 35 51 66 77 87 96 110 126 140 152 256	32.2021	55 62 86 99 128 147 161 171 182 194 211 231	26 45 75 107 128 143 156 168 182 197 231 242	60 62 111 114 142 167 177 183 184 229 250 254	36.6335
Test3												
4	38 75 104 144	44 64 100 142	83 134 172 193	24.3252	40 71 106 152	42 74 106 144	73 120 156 187	24.8115	40 70 105 151	43 76 108 147	72 119 155 187	27.3306
8	25 50 66 82 99 117 141 168	27 40 56 84 91 107 134 158	52 85 120 144 167 185 200 207	30.4331	26 46 63 81 101 123 148 178	18 27 61 155 164 171 176 207	41 65 93 120 142 162 184 210	27.4156	26 46 63 81 102 123 148 178	32 56 75 92 112 135 163 256	46 73 103 131 152 169 188 211	31.4123
10	30 37 44 55 79 94 97 116 149 172	26 46 64 81 99 129 140 143 151 160	42 60 81 91 123 143 159 165 186 216	32.1253	10 14 112 121 153 157 203 208 215 237	1 12 26 43 61 77 94 114 136 164	37 56 77 101 124 142 158 174 191 214	31.6257	23 39 52 67 84 103 122 143 165 183	21 40 58 72 85 99 115 133 152 174	13 42 62 86 112 135 154 171 190 213	35.1477
12	29 46 51 54 73 88 97 97 121 134 134 172	19 38 45 61 78 94 104 109 133 141 157 162	48 85 104 125 148 152 166 172 178 200 207 21	35.6188	1 15 29 44 56 69 85 103 123 145 173 202	18 32 47 61 74 86 98 114 132 152 178 256	21 39 58 80 103 124 140 154 167 181 196 216	32.6044	16 31 44 56 69 84 99 118 136 159 180 197	17 33 49 62 76 90 106 123 140 160 177 199	19 37 51 66 86 107 127 145 160 176 193 216	38.6449
Test4												
4	108 142 177 199	75 113 151 179	78 109 132 153	24.4314	13 21 41 53	89 121 153 183	71 101 129 156	22.3045	103 136 167 195	93 126 157 185	86 120 151 231	26.5114
8	73 109 119 136 165 178 186 204	80 87 92 95 129 156 177 194	51 85 98 112 127 139 144 168	28.5112	51 82 102 121 141 164 185 205	39 73 95 114 135 154 174 193	40 62 78 95 114 131 148 168	28.1244	51 84 108 129 149 170 189 208	66 98 130 159 186 244 253 253	48 71 89 106 121 136 152 171	30.6814
10	78 87 89 105 127 143 153 168 186 208	61 91 111 126 131 151 170 173 185 197	54 84 89 99 113 132 146 147 156 166	31.3547	60 78 96 113 130 148 165 180 194 210	44 66 85 101 113 128 144 159 176 196	38 61 78 93 107 120 133 145 159 175	31.1274	44 45 71 89 108 128 149 170 189 208	65 85 102 118 135 152 167 182 199 249	39 62 78 93 107 122 135 149 166 190	34.6817
12	77 98 102 103 112 129 152 153 173 183 196 209	73 92 102 115 131 134 134 152 160 174 199 199	56 57 72 87 104 110 114 118 130 148 159 176	35.8573	20 38 167 167 193 198 219 219 221 225 235 254	1 49 69 83 95 106 123 140 156 171 186 202	39 54 68 81 94 106 117 127 137 148 161 177	35.1106	33 42 57 80 99 117 134 151 167 182 195 211	1 6 33 59 81 100 119 136 153 168 183 200	26 47 64 79 93 105 117 128 138 149 162 176	38.3745
Test5												
4	54 99 182 195	59 101 147 207	50 90 135 194	21.5258	51 91 147 212	50 82 135 206	51 81 132 205	21.3287	51 91 147 213	50 82 135 206	51 81 132 205	18.2217
8	36 49 56 88 135 144 195 225	47 74 99 106 146 152 219 228	45 66 70 88 112 150 214 224	26.7221	35 50 70 94 119 152 189 230	42 55 71 92 118 149 185 228	40 52 64 82 108 142 182 226	24.8202	34 49 69 96 122 154 188 226	42 55 72 93 119 150 186 227	41 54 67 85 110 142 180 225	30.7091
10	35 38 44 58 78 113 119 158 168 225	43 58 86 91 96 119 131 147 200 214	19 47 59 62 64 77 119 133 187 225	31.4014	31 40 51 68 89 113 140 170 200 233	37 46 56 69 87 109 139 171 204 237	39 50 60 74 92 117 143 171 200 228	30.5911	32 42 55 72 95 118 144 172 200 233	39 49 61 77 97 120 144 166 191 226	9 40 52 65 82 107 138 175 205 231	34.4815
12	35 51 74 97 127 131 151 163 187 200 224 229	41 45 56 66 94 132 160 191 194 211 214 226	39 53 68 81 92 102 115 135 149 208 210 228	33.9251	13 30 39 50 65 82 102 122 146 171 199 233	35 41 49 57 66 80 96 116 138 162 192 229	36 44 52 60 69 82 98 120 147 177 208 235	32.4124	30 39 48 61 77 94 111 126 140 162 191 228	38 47 56 67 83 102 125 151 173 194 214 233	36 44 53 62 73 89 110 137 167 197 223 237	37.8463
Test6												
4	57 90 113 145	88 159 176 182	64 117 132 157	23.7211	57 105 132 152	57 105 132 152	57 107 136 156	21.7249	54 82 119 191	57 83 108 175	32 56 80 141	26.7357
8	37 43 50 92 115 136 145 160	50 82 106 122 136 154 172 194	27 57 101 114 135 140 153 168	26.7189	70 88 168 186 196 196 221 227	32 50 66 95 127 151 171 189	3 53 82 107 125 140 157 256	25.9111	39 61 81 106 137 180 201 225	40 58 76 91 108 138 176 218	24 42 60 78 104 134 167 191	30.7447
10	31 71 81 108 117 133 143 146 155 171	37 45 46 87 107 144 159 178 188 193	23 33 57 65 72 92 124 137 156 165	31.5353	20 40 63 87 104 117 129 140 151 161	22 50 54 90 98 138 205 206 209 215	9 17 31 51 78 101 118 133 147 162	31.2254	30 46 61 76 93 116 142 164 180 217	36 52 65 78 91 103 120 149 170 215	19 35 50 66 83 109 137 148 162 179	34.5216
12	27 33 71 88 102 119 123 137 151 157 181 206	38 65 84 99 100 116 127 127 142 166 182 194	22 31 53 62 65 72 84 116 130 147 151 165	36.0584	18 34 56 79 97 111 123 133 143 152 162 195	38 63 87 109 122 136 150 163 176 191 217 226	1 6 19 38 61 88 109 122 133 143 154 166	32.0119	25 39 53 66 78 91 109 132 165 199 203 224	39 56 74 89 106 136 175 182 187 192 201 217	13 25 36 47 59 70 82 99 114 141 167 193	38.4324
Leaf1												
4	79 95 114 190	96 113 132 199	48 63 81 173	22.5158	108 120 132 146	132 145 156 170	73 85 97 112	21.3547	79 94 113 189	95 112 131 199	48 63 81 169	19.2117
8	70 81 90	85 96	42 53 64	26.1284	41 44 78 96	123 131	32 44 80 142	22.8202	69 81 91 102	84 95 104	40 50 60 73	31.7131

TABLE 3. *(Continued.)* **The optimal fitness value and threshold value of each algorithm under Tsallis.**

	100 112 129 178 209	106 115 126 141 165 208	76 95 187 194 248		121 152 180 190	138 145 152 159 167 180	157 172 177 206		117 152 198 222	113 123 136 162 226	91 113 157 160	
10	67 76 84 91 99 108 120 141 214 254	55 84 97 105 114 128 139 141 162 229	11 38 48 55 64 74 88 112 140 182	31.5717	48 106 141 177 178 204 207 221 224 239	8 10 43 43 97 103 112 122 197 225	31 49 66 66 97 123 171 175 211 246	25.1813	69 80 90 100 114 140 174 216 224 225	84 96 107 118 132 157 190 192 192 205	42 53 64 76 94 165 170 184 238 249	35.4535
12	63 72 79 86 92 99 106 115 130 155 204 227	5 10 16 52 60 78 102 116 119 126 157 218	35 44 51 59 66 74 80 95 179 183 186 240	35.1351	12 23 35 40 40 98 111 184 217 241 243 246	20 35 43 48 54 74 77 86 105 108 111 130	20 34 39 45 50 92 97 164 236 243 244 246	31.4008	26 32 47 53 72 73 88 98 161 176 177 200	26 32 47 53 72 73 88 98 161 176 177 200	11 42 54 66 82 113 166 186 211 211 220 240	38.8573
Leaf2												
4	108 120 132 146	74 86 98 112	108 120 132 147	25.8591	79 95 114 185	96 113 132 199	48 63 81 201	21.1124	107 120 132 146	132 145 156 170	74 86 97 112	27.7456
8	67 78 87 96 106 118 144 191	44 68 77 85 92 100 109 122	101 109 116 124 131 137 147 157	27.2189	69 79 87 95 104 116 135 205	55 87 99 110 122 139 178 222	39 49 59 69 84 120 192 209	24.4221	22 33 68 83 130 174 176 251	31 71 86 112 116 167 177 181	54 65 108 109 192 199 208 221	31.7119
10	25 68 82 82 88 131 158 177 193 201	64 72 78 85 91 98 104 111 116 129	99 106 112 118 123 129 134 141 149 161	31.3223	67 77 86 94 103 114 135 195 213 229	80 90 99 106 113 122 135 154 159 217	42 53 63 74 91 156 158 211 211 245	30.2011	27 32 59 74 76 112 122 228 242 250	39 40 145 159 180 182 198 220 236 251	2 8 90 126 192 207 209 213 237 248	35.5154
12	21 46 113 125 145 148 160 174 184 186 211 223	22 33 62 146 150 177 180 205 213 225 244 246	53 60 76 157 168 187 201 214 227 234 239 251	35.1854	62 71 78 85 92 99 107 116 131 158 186 193	33 34 51 59 85 93 113 119 134 217 228 251	39 49 58 67 78 96 136 148 161 239 241 247	32.0149	9 25 84 86 91 101 101 104 135 190 191 253	13 43 47 74 119 120 133 138 149 162 192 222	6 45 45 87 87 108 138 188 217 221 242 255	39.2224

TABLE 4. **The PSNR and FSIM of each algorithm under Tsallis.**

K	GOA		WOA		FPA		PSO		BA		MGOA	
	PSNR	FSIM	PSNR	FSIM	PSNR	FSIM	PSNR	FSIM	PSNR	FSIM	PSNR	FSIM
Test1												
4	17.9478	0.8658	17.9744	0.8654	17.5541	0.7738	16.9008	0.7734	17.8235	0.8679	17.9279	0.8656
8	22.8018	0.9469	23.2061	0.9463	21.3791	0.8774	21.3832	0.8814	20.9239	0.8788	25.8755	0.9474
10	24.4042	0.9615	25.1398	0.9606	22.5334	0.9073	22.6249	0.8982	22.0579	0.8811	26.9691	0.9623
12	25.4411	0.9693	27.1819	0.9611	24.2467	0.9162	23.2354	0.9090	25.4246	0.9721	27.8297	0.9782
Test2												
4	22.4952	0.8292	22.3392	0.8277	22.3470	0.8280	22.3044	0.8380	19.9078	0.7768	22.4671	0.8292
8	27.0545	0.8819	27.8419	0.9012	28.1530	0.8990	26.4004	0.9039	22.2960	0.8307	28.0998	0.8921
10	27.4640	0.8933	29.5160	0.9198	28.7455	0.9107	27.6596	0.9219	30.1602	0.9256	28.6312	0.9465
12	30.3517	0.9016	30.7578	0.9309	31.1217	0.9360	29.2649	0.9344	28.7392	0.9115	29.3186	0.9521
Test3												
4	20.9266	0.7750	20.9181	0.7749	20.9579	0.7750	20.5956	0.6997	20.9783	0.7750	20.9411	0.7752
8	25.5619	0.8472	26.1558	0.8548	25.3810	0.8438	24.8984	0.8015	22.7276	0.8337	25.5809	0.8468
10	27.4053	0.8734	27.2407	0.8703	26.5908	0.8643	26.0738	0.8212	22.9602	0.8186	27.7349	0.8791
12	28.7212	0.8921	29.1073	0.8982	28.5093	0.8935	26.5495	0.8351	28.9372	0.8945	29.0314	0.9013
Test4												
4	18.9626	0.7831	18.9191	0.7835	18.3870	0.7677	19.1694	0.7289	15.6841	0.6772	18.9221	0.8012
8	25.3749	0.8512	26.3351	0.9186	24.0158	0.8850	23.8865	0.8425	26.5919	0.9210	23.9925	0.8793
10	27.6053	0.9234	28.1813	0.9443	25.8145	0.9118	25.7121	0.8793	28.4094	0.9464	26.9317	0.9321
12	29.4653	0.9442	29.5916	0.9573	27.9701	0.9419	25.6696	0.8875	23.8150	0.8296	29.2726	0.9621
Test5												
4	18.1592	0.7571	18.1830	0.7560	18.1572	0.7571	17.3974	0.3912	18.2168	0.7569	18.2209	0.7569
8	22.7119	0.8585	22.4799	0.8569	23.0034	0.8583	20.9141	0.5870	22.6567	0.8583	22.5721	0.8585
10	23.8266	0.8806	24.4292	0.8854	23.5906	0.8748	22.6229	0.6898	24.1356	0.8840	24.1841	0.8836
12	24.9569	0.8975	24.9303	0.9005	25.5315	0.9002	22.9635	0.6862	25.9704	0.9065	24.7992	0.9119
Test6												
4	21.1873	0.8155	22.4419	0.8542	21.1909	0.8160	21.0133	0.7716	22.5738	0.8542	21.2549	0.8182
8	26.4465	0.8941	28.0235	0.9405	25.7154	0.9060	26.3730	0.8745	22.5706	0.8094	25.7571	0.9045
10	28.0764	0.9242	29.3653	0.9557	27.9539	0.9342	27.0074	0.8865	25.4198	0.8494	27.6659	0.9421
12	29.3102	0.9453	30.4888	0.9633	29.1162	0.9418	28.0076	0.8944	30.0112	0.9362	28.6685	0.9691
Leaf1												
4	16.9471	0.7963	17.2417	0.7913	16.8802	0.8247	16.9702	0.7810	16.8993	0.7749	16.8993	0.7549
8	21.1124	0.8868	21.7189	0.8760	21.7330	0.8973	21.8629	0.8779	21.9861	0.8757	21.9861	0.8952
10	22.5191	0.8928	23.2649	0.8966	26.2076	0.9377	22.2724	0.8883	24.8468	0.9526	24.8468	0.9621
12	24.9018	0.9262	24.2784	0.9267	27.8514	0.9616	24.9449	0.9316	26.6965	0.9639	26.6965	0.9734
Leaf2												
4	16.6919	0.7505	16.6953	0.7488	16.5371	0.7478	16.6917	0.7396	16.6912	0.7398	17.3023	0.7758
8	19.5653	0.7712	17.1414	0.7444	16.4690	0.7612	17.8660	0.8831	17.2795	0.8600	23.7610	0.9330
10	22.3245	0.8212	21.5455	0.8017	21.7875	0.7891	22.1557	0.8767	19.4523	0.8895	23.8966	0.9130
12	24.6540	0.8732	23.5612	0.8491	22.0913	0.8012	23.0715	0.9549	22.9223	0.9369	25.6974	0.9573

of flowers; The Bat Algorithm (BA) [38]–[42] is an efficient global search method, which can search the optimal solution by iteration and near optimal solution in flight to produce local data processing, strengthen the local search ability; The Whale Optimization Algorithm (WOA) [43]–[45] proposed by Mirjalili in 2016 is to simulate the predation of humpback whales. The algorithm has strong capability in the global search and local optimization. Although there are differences between evolutionary optimization and swarm optimization, the common denominator is the ability to find the optimal value of a restricted domain [46]. Although each algorithm has its own advantages, No-Free-Lunch [47] has proved that no algorithm can solve all optimization problems.

Therefore, there is no perfect optimization algorithm and the optimization algorithm should be improved to better solve engineering problems. Many scholars study Levy flight improve optimization algorithm. Levy flight is a random walk strategy whose step length obeys the Levy distribution and it can maximize the efficiency of resource searches in uncertain environments. Haklı and H. Uğuz [48] proposed the PSO algorithm which combined with Levy flight. The method can overcome the problems as being trapped in local minima due to premature convergence and weakness

TABLE 5. The optimal fitness value and threshold value of each algorithm under Otsu.

K	GOA				WOA				FPA			
	R	G	B	F	R	G	B	F	R	G	B	F
Test1												
4	39 89	46 89	46 84		39 89 161	46 89	47 85 131		39 88 159	45 88	47 85	
	160 226	137 208	130 204	8256.5271	227	137 208	206	8256.6119	225	136 208	132 206	8256.5411
8	11 42 79	26 49	26 46 66		20 44 70	28 50 72				27 49	26 45 63	
	83 93	71 95	87 110		98 131	95 119	27 47 67		20 44 70 97	71 94	82 103	
	116 120	120 148	137 173		175 214	146 180	88 111 139		130 174	119 146	131 168	
	250	183 229	224	8359.0255	239	227	175 223	8363.8067	213 238	181 226	222	8363.4207
10	13 31 51	21 38	100 102		14 33 51	23 42 61	25 42 58		15 35 56 79	22 41	19 34 50	
	71 94	56 74	152 162		73 97 124	79 100	74 91 110		103 132	60 79	67 86	
	122 154	94 113	185 211		158 192	119 139	131 158		168 200	99 121	105 127	
	189 218	133 157	214 216	8372.2668	219 241	162 186	193 232	8378.2028	222 242	143 167	155 192	8377.6654
	241	186 227	252 255			226				197 232	228	
12	11 27 46	87 89	14 18 25							18 32		
	64 83	98 158	53 75		14 31 50	1 20 35	1 23 38 53		7 18 30 44	49 65	17 28 41	
	103 124	178 207	173 176		69 87 107	52 68 85	68 84 99		60 77 97	82 99	55 69 84	
	151 179	230 239	191 200		129 152	104 124	116 135		119 146	117 135	100 118	
	203 222	243 245	202 210		177 202	146 172	159 189		183 218	155 177	138 163	
	242	246 250	216	8376.2232	221 241	203 236	230	8384.2418	241	202 236	195 234	8385.4179
Test2												
4	32 68	106 141	53 87		113 161	106 141	64 94 120		113 160	16 51	63 94	
	128 245	162 185	114 141	646.10259	181 210	162 185	144	655.78846	181 209	216 227	120 144	647.0593
8	69 113	20 38	1 14 42		47 64 89	18 69	44 76 95		54 106 141	70 109	37 63 81	
	144 160	58 83	49 85		143 164	113 137	115 134		157 170	131 145	95 111	
	171 182	86 219	166 172		178 194	153 167	149 171		182 198	157 169	127 140	
	194 217	222 231	174	677.64718	219	184 208	178	686.20867	222	185 208	152	694.24362
10	35 69	44 74	44 74		45 52 68	35 48 83			72 110 135	57 60	2 26 59	
	106 138	108 127	108 127		103 144	110 130	4 37 66 84		151 163	65 75	78 93	
	156 169	140 151	140 151		160 172	145 157	99 117 135		173 183	83 99	109 125	
	180 191	161 170	161 170		184 200	169 186	149 174		195 212	175 201	139 151	
	208 230	180 198	180 198	694.30936	223	215	188	694.36193	232	234 237	256	694.19962
12	31 31 64	11 60	11 60 91		32 36 41	27 42 55			47 71 101	57 82		
	85 125	91 111	111 125		72 106	56 81	27 42 55		125 142	102 118	14 36 58	
	146 159	125 137	137 147		140 157	109 130	56 81 109		154 164	129 140	73 85 96	
	170 181	147 155	155 163		169 180	145 157	130 145		173 183	149 157	108 122	
	194 213	163 172	172 184		192 211	169 183	157 169		194 211	165 174	133 144	
	233	184 203	203	702.24755	231	206	183 206	698.53433	230	189 224	153 175	703.74901
Test3												
4	39 68	43 76	73 120		40 70 106	43 77	71 119 155		113 160	16 51	63 94	
	104 149	108 147	156 187	1901.6364	152	109 148	187	1901.8564	181 209	216 227	120 144	1942.8564
8	27 46 63	19 39	44 68 96		28 47 64	24 47 69	2 49 80		54 106 141	70 109	37 63 81	
	81 100	59 75	124 145		82 101	90 114	111 138		157 170	131 145	95 111	
	121 145	92 112	163 183		122 146	141 170	160 183		182 198	157 169	127 140	
	176	134 163	209	1983.8059	175	186	209	1977.0473	222	185 208	152	1997.0473
10		19 36	33 51 71						72 110 135	57 60	2 26 59	
	21 37 50	53 67	96 121		19 34 49	21 40 58	35 51 71		151 163	65 75	78 93	
	62 76 92	79 93	141 157		63 79 96	71 84 98	95 121 143		173 183	83 99	109 125	
	110 130	106 123	174 191		116 137	115 132	162 183		195 212	175 201	139 151	
	154 182	141 165	214	1995.5656	165 195	151 175	208 251	1992.4759	232	234 237	256	2001.4759
12		4 14 25	32 46 66						47 71 101	57 82		
	17 31 44	41 56	87 109		16 31 45	1 23 41	32 50 71		125 142	102 118	14 36 58	
	55 68 84	69 80	127 143		56 68 82	59 72 84	93 115 133		154 164	129 140	73 85 96	
	100 117	94 110	157 170		97 113	97 112	147 160		173 183	149 157	108 122	
	136 159	131 154	184 198		132 154	128 146	172 185		194 211	165 174	133 144	
	184 216	180	217	2001.2325	175 195	169 225	199 218	2001.6745	230	189 224	153 175	2015.6745
Test4												
4	103 136	3 17 18	9 33 56		103 136	93 126	75 105 131		102 135	93 126	13 42	
	167 195	45	57	1353.3071	167 195	157 185	157	1391.8873	166 195	157 185	172 191	1368.7432
8	12 76	59 79	2 22 133		48 73 100	66 88			58 76 97	58 84	43 65 83	
	102 122	96 114	137 203		122 145	106 124	1 52 78 99		114 132	105 125	100 118	
	144 166	135 155	211 214		167 187	142 161	117 133		153 177	144 162	134 150	
	186 206	175 195	217	1440.9301	207	179 198	150 169	1446.8713	200	178 196	170	1446.2636
10	52 73 92	47 56	32 54 69		11 11 69	68 90			51 65 90	6 57 79	39 54 68	
	109 127	78 96	84 100		96 120	109 128	1 48 72 90		110 128	96 113	84 98	
	145 162	115 132	116 131		144 165	146 163	106 121		146 163	132 150	113 126	
	179 194	149 165	144 159		185 204	180 198	136 152		178 193	166 181	139 154	
	210	180 197	181	1457.4504	227	245 249	171 236	1449.8233	209	199	172	1457.3396
12	59 76 89	30 57			45 66 86	1 1 58			60 83 101	31 33		
	103 116	73 88	33 44 59		103 118	79 96	1 20 40 60		116 131	47 71	24 24 28	
	129 143	101 115	73 87 99		134 149	112 128	76 92 107		144 157	72 83	56 59 60	
	158 172	130 146	111 124		163 177	144 159	121 133		169 181	162 193	75 81	
	186 199	161 176	135 147		190 203	173 186	146 160		192 203	203 223	121 209	
	214	190 205	162 179	1464.6615	217	202	178	1462.501	215	227 245	220 220	1453.8639
Test5												
4	32 68	106 141	53 87		51 91 148	51 83	51 81 132		51 91 147	51 83	51 81	
	128 245	162 185	114 141	3314.9128	214	135 206	205	3334.9128	213	135 205	132 205	13334.901
8	69 113	20 38	1 14 42		34 50 70	42 55 71				41 54	13 13 99	
	144 160	58 83	49 85		94 120	92 118	41 54 67		34 49 68 92	70 92	142 187	
	171 182	86 219	166 172		151 188	148 184	85 109 141		118 150	119 150	223 226	
	194 217	222 231	174	3406.4766	229	227	179 224	3411.4766	188 228	185 226	251	3397.3674
10	35 69	44 74	44 74		33 44 59	39 50 64				38 48	38 49 59	
	106 138	108 127	108 127		81 106	84 109	39 50 60		30 41 54 71	58 70	70 86	
	156 169	140 151	140 151		132 159	143 173	72 89 112		93 115 140	85 102	106 130	
	180 191	161 170	161 170		185 204	195 208	140 167		164 193	124 150	159 191	
	208 230	180 198	180 198	3411.1442	233	230	192 228	3419.1542	229	182 226	231	3420.772
12	31 31 64	11 60	11 60 91			38 46 55				35 44		
	85 125	91 111	111 125		31 40 50	67 83	37 46 55		29 38 48 60	53 63	35 42 49	
	146 159	125 137	137 147		62 79 96	102 125	64 77 94		78 99 121	77 95	56 65 77	
	170 181	147 155	155 163		113 130	150 173	114 137		144 168	114 136	94 115	
	194 213	163 172	172 184		149 173	199 213	163 182		194 216	161 185	141 165	
	233	184 203	203	3416.5455	198 232	236	202 231	3426.5465	240	211 236	193 232	3426.687
Test6												
4	16 145	19 50	58 109		59 106	81 135	58 109 137			48 100	50 100	
	189 193	189 240	137 157	1647.2302	133 153	165 186	157	1660.0172	6 15 18 155	148 176	134 155	1652.2628
8	22 37 62	41 68	6 27 34		27 55 83	39 66 95	24 53 87			23 46	18 35 65	
	88 108	97 126	112 173		105 122	125 145	114 132		21 39 61 86	70 101	99 121	
	125 142	146 162	182 191		136 149	162 177	146 161		106 124	134 156	135 147	
	158	176 191	201	1703.7498	161	191	197	1709.8929	139 156	174 190	161	1707.9906
10	25 46 63	21 39	21 39 64			42 73				28 42	22 45 75	
	84 105	64 93	93 120		1 27 52 77	106 126	16 34 62		18 34 54 72	59 77	98 113	
	121 136	120 142	142 156		101 120	144 160	92 113 128		93 110 124	106 131	126 136	
	148 160	156 170	170 183		135 148	175 190	140 150		137 148	149 163	146 155	
	256	183 196	196	1711.0219	160 230	249 249	162 179	1713.2302	159	177 191	167	1717.6573
12	18 30 45	12 47	12 47 48							6 36 65	15 27 41	
	65 87	48 51	51 61		1 24 44 66	15 25 29	19 38 63		15 30 51 74	94 119	59 83	
	104 118	61 161	161 162		89 106	34 45 61	87 105 117		94 109 122	137 149	102 116	
	130 140	162 165	165 201		121 133	86 118	126 135		134 145	160 170	128 138	
	150 161	201 229	229 241		144 154	143 161	144 152		155 165	179 188	148 158	
	230	241 247	247	1712.5531	164 199	176 191	160 170	1718.5778	228	198	169	1721.9268

TABLE 5. ***(Continued.)*** **The optimal fitness value and threshold value of each algorithm under Otsu.**

Leaf1												
4	79 95	96 116	48 63 81		79 94 113	95 112	52 72 160		79 94 113	96 113	48 63 81	
	114 190	148 210	187	1825.2078	190	131 199	247	1821.4726	188	132 199	173	1827.1329
8	68 78 87	84 95			72 84 95	86 96				90 104	45 58 71	
	95 104	104 113	36 46 55		107 127	105 114	44 57 70		69 79 88 96	117 132	89 192	
	117 141	123 136	65 76 91		218 229	124 137	87 228 228		105 115	175 220	211 240	
	204	156 207	99 169	1850.2119	250	162 239	233 238	1844.4818	130 189	222 229	249	1843.3636
10	67 78 87	7 75 94				85 97				85 96		
	96 106	95 102	37 47 56		41 68 77	108 119	37 47 56		69 80 89 99	106 117	40 49 58	
	118 146	111 219	66 78 97		85 92 101	133 161	65 75 88		110 126	130 148	67 78 97	
	187 188	226 227	138 181		112 129	223 224	94 97 153		170 175	188 189	148 179	
	196	230	184 215	1850.4216	171 198	228 230	200	1849.6088	176 194	232 233	181 245	1849.1687
12	67 76 84	65 85	40 50 59			85 95				81 90		
	91 99	95 104	68 80		63 72 79	104 113	34 43 50		69 79 87 95	98 106	37 45 53	
	108 120	112 122	101 163		87 94 102	123 135	57 64 72		104 115	114 123	61 69 79	
	146 226	133 152	206 234		112 125	156 216	82 94 117		133 197	133 148	94 110	
	233 240	195 215	243 247		159 182	235 241	179 186		202 203	180 183	157 157	
	243	232 244	247	1851.2717	194 209	252 253	190	1852.7437	230 241	241 248	186 240	1852.7794
Leaf2												
4	21 112	132 144	38 140		107 120	131 143	98 150 153		85 152 170	132 144	74 86 97	
	179 196	155 169	171 178	201.7214	131 146	155 169	173	201.747	243	156 170	111	201.77025
8	45 75 84	21 61	3 50 93		98 106	3 27 84	23 56 72		54 100 138	19 54	24 87	
	101 127	90 101	102 111		113 120	91 116	112 119		153 165	66 69	139 160	
	156 204	128 169	126 182		127 135	135 182	125 165		219 231	105 132	187 207	
	206	238 245	234	202.1214	144 157	212	244	202.1214	232	193 222	210 219	202.12357
10	9 41 71	1 24 29			9 32 77	1 69 88	140 141			40 85	40 54 61	
	116 140	55 136	12 19 39		104 111	88 90	145 156		16 76 81 97	103 108	87 101	
	148 192	144 151	42 72 85		141 151	100 182	164 167		100 158	118 138	168 179	
	211 219	184 186	160 168		172 184	221 239	183 200		164 201	162 165	243 249	
	247	188	177 233	203.12456	234	250	207 240	203.12456	246 251	191 216	251	204.12475
12		30 53			32 56 71	20 34 77				21 41	9 14 58	
	19 27 36	78 83	1 4 11		102 108	81 99	5 11 45 70		6 24 30 90	74 93	97 100	
	57 59 92	84 119	48 75 76		134 155	146 156	79 91 116		120 150	142 144	100 113	
	125 137	137 151	104 113		156 183	177 197	124 124		161 194	151 151	148 197	
	151 153	166 217	151 167		235 238	230 233	180 227		217 220	153 155	222 234	
	157 209	225 234	237 247	206.21544	243	253	238	207.21544	235 243	177 238	247	208.21476

FIGURE 5. **The Tsallis results of image test1.**

of global search capability. Amirsadri *et al.* [49] proposed a new algorithm benefits from simultaneously local and global search, eliminating the problem of getting stuck in local optima. The method using Levy flight improves the gray wolf optimizer (GWO). The modified algorithm balances the exploration and exploitation of the GWO.

TABLE 6. The optimal fitness value and threshold value of each algorithm under Otsu.

K	PSO				BA				MGOA			
	R	G	B	F	R	G	B	F	R	G	B	F
Test1												
4	39 89 160 226	46 89 137 207	48 86 132 206	8256.6026	39 89 159 226	45 88 136 208	47 85 131 206	8256.5591	40 90 161 227	44 91 167 221	43 91 164 221	8256.638
8	18 40 65 92 123 164 204 235	28 51 72 95 119 146 180 226	29 49 69 90 114 142 179 226	8362.9878	20 44 70 97 130 176 216 240	25 47 68 90 115 143 179 226	29 49 69 90 113 140 176 226	8363.7497	14 40 67 99 134 175 217 240	28 51 73 97 121 149 183 229	18 39 60 83 108 135 178 226	8361.9012
10	13 31 52 73 96 123 159 192 217 240	16 26 130 135 184 190 201 211 242 249	21 36 50 65 84 105 129 159 195 234	8373.2545	12 28 48 69 92 119 149 187 218 241	21 39 57 75 94 116 139 166 199 234	15 27 42 59 78 98 121 152 191 234	8377.0568	15 31 50 71 95 121 153 188 219 242	24 43 61 79 98 119 141 167 198 235	6 29 47 65 84 104 127 159 193 234	8377.2386
12	10 24 40 57 76 95 114 137 159 191 217 240	19 33 48 62 77 94 112 131 153 177 206 237	17 31 45 59 73 90 108 129 154 183 212 238	8385.9662	12 27 44 61 79 100 124 152 184 210 228 245	16 31 47 63 79 96 115 135 156 180 206 233	17 29 42 57 72 89 107 127 151 179 210 237	8386.3098	13 29 44 64 83 106 125 156 185 211 229 245	21 38 53 68 83 99 116 134 154 177 205 239	21 26 36 48 63 80 97 117 138 166 201 235	8385.9828
Test2												
4	112 161 181 210	103 139 160 184	59 90 115 143	655.38229	33 37 95 120	11 49 50 244	16 18 24 184	604.86868	114 161 181 210	106 141 162 186	63 94 120 144	655.8005
8	41 43 85 85 125 146 233 246	13 77 82 89 193 216 236 236	32 56 71 86 102 120 137 150	673.60138	73 109 143 158 171 183 199 222	68 101 121 137 149 160 171 189	42 72 89 105 123 138 151 256	693.09591	81 106 146 161 173 185 203 225	50 95 122 138 151 162 172 191	26 42 70 86 100 115 133 151	692.73275
10	77 117 140 154 165 174 183 195 211 231	61 74 100 104 113 181 186 214 229 248	18 30 48 72 88 104 122 138 151 256	693.45932	35 72 97 136 154 167 178 190 206 228	48 87 113 130 143 153 163 172 187 214	33 61 77 89 102 116 130 142 152 173	699.49055	36 70 109 146 161 172 182 194 212 233	62 72 77 114 134 147 159 170 186 209	19 53 69 82 98 113 128 138 149 157	698.46566
12	31 65 102 129 148 161 171 181 192 205 218 233	46 77 90 110 128 140 150 160 170 183 201 230	6 11 44 56 69 82 106 108 153 161 184 196	700.53076	64 104 131 145 157 166 174 183 193 206 221 235	15 22 45 81 83 103 174 176 189 204 209 218	1 25 45 46 93 131 156 167 200 217 245 255	705.49518	58 86 103 138 153 164 173 182 193 206 224 239	63 94 112 128 139 148 157 165 173 185 203 218	5 26 57 74 85 94 105 116 130 140 149 156	704.36212
Test3												
4	40 70 105 151	43 76 108 146	67 114 152 185	1901.4771	40 70 105 151	43 76 108 147	71 117 154 186	1901.75	40 71 106 152	43 77 109 148	71 119 156 188	1901.8557
8	28 47 64 82 102 124 150 180	22 41 60 76 93 113 135 163	35 56 79 105 134 158 180 204	1981.8197	23 39 54 72 95 119 151 189	23 42 60 76 93 113 135 163	42 66 95 121 141 161 181 207	1982.1471	28 48 65 83 103 124 149 178	24 42 61 77 94 114 136 164	45 70 100 131 153 171 190 214	1984.5747
10	23 39 53 67 82 98 116 136 160 187	1 22 40 57 71 85 101 120 141 167	39 61 84 109 130 148 163 178 195 215	1995.2676	22 38 52 66 82 99 116 136 160 186	19 35 51 64 77 91 108 127 148 173	35 39 50 73 79 83 102 208 242 250	1988.4151	22 41 55 70 85 101 118 137 161 188	21 28 44 61 76 91 108 126 146 170	9 45 69 95 121 142 159 175 193 214	1994.2647
12	16 29 44 57 72 89 110 129 148 169 185 256	15 27 41 55 67 78 91 105 122 135 151 175	24 54 55 60 61 101 102 105 118 129 210 211	1998.499	17 31 45 56 69 85 102 119 138 158 177 195	13 24 39 54 66 78 91 108 124 141 161 183	26 39 55 74 98 120 138 153 166 180 195 214	2002.2347	20 35 46 57 69 83 97 111 126 143 165 189	20 36 48 61 73 85 98 112 129 144 163 185	13 36 59 79 100 123 142 156 170 184 199 219	2002.9132
Test4												
4	205 208 221 221	54 223 235 244	73 104 130 156	1337.4743	40 46 91 104	6 17 47 219	7 7 184 217	1328.3622	103 136 167 195	16 103 143 178	73 104 131 157	1377.1201
8	19 32 91 111 127 208 226 237	65 86 105 124 142 160 176 196	50 71 89 106 121 136 152 171	1445.7059	40 76 97 118 138 160 182 204	60 81 99 118 139 159 178 197	27 42 64 83 102 122 141 163	1443.5726	25 82 108 130 151 171 189 208	68 93 112 131 149 166 182 199	50 73 91 108 123 137 153 172	1449.3992
10	70 93 110 126 142 157 172 185 198 213	21 37 72 73 149 161 209 212 222 224	41 62 80 95 109 122 134 146 159 175	1455.2236	60 78 95 112 130 147 164 179 193 210	32 42 53 76 91 194 231 240 241 251	45 67 85 101 116 129 141 155 173 256	1448.502	31 51 87 108 126 146 164 180 194 211	17 21 66 91 111 131 148 165 181 199	44 66 82 96 110 123 135 146 159 176	1455.7686
12	30 48 71 93 110 126 141 156 171 185 198 213	1 33 54 74 93 109 123 139 154 168 182 199	42 62 76 90 103 115 126 136 147 160 176 227	1462.4138	53 74 90 103 116 129 143 158 171 184 196 210	6 51 71 82 93 106 121 137 153 167 181 198	1 38 57 73 86 99 111 124 135 147 161 177	1463.4149	15 35 53 83 104 121 139 155 171 185 199 215	34 59 83 100 117 133 149 163 173 184 196 210	11 33 56 74 87 101 114 126 137 148 161 179	1462.5587
Test5												
4	51 91 148 214	23 40 199 252	51 81 132 205	3329.6811	50 89 146 213	51 83 136 206	51 81 131 204	3334.8426	51 91 148 214	51 83 136 206	51 81 132 205	3334.9229
8	34 47 65 89 115 147 184 228	41 54 69 90 116 147 183 226	41 53 66 84 108 138 176 222	3411.1733	34 49 68 92 118 150 188 229	40 53 67 87 113 146 182 226	40 51 63 81 107 140 179 226	3411.1353	35 49 69 93 119 152 190 230	41 56 76 99 132 161 192 231	41 54 69 88 111 143 182 228	3410.9039
10	16 137 141 142 144 181 190 195 239 244	39 50 61 77 96 119 146 174 201 233	19 67 68 140 180 190 211 244 245 255	3408.2907	31 41 53 70 91 113 137 167 201 235	38 49 60 75 95 118 146 175 200 229	39 50 60 72 88 109 135 163 195 231	3421.3767	32 43 55 72 93 116 141 172 203 236	39 50 61 76 95 118 144 174 203 235	15 41 52 62 75 94 117 149 186 229	3420.4357
12	30 39 48 59 75 95 115 136 162 185 207 235	37 47 57 69 85 106 128 150 174 193 209 235	36 45 55 66 81 101 125 149 172 191 211 236	3426.2966	25 32 41 53 69 89 112 134 157 182 211 236	36 45 53 63 77 95 115 140 166 187 210 235	9 38 48 59 71 90 112 134 152 174 193 228	3425.0311	30 39 49 61 80 97 118 138 162 185 210 240	38 47 58 68 82 98 115 136 158 182 209 239	37 46 56 65 79 96 117 140 167 193 205 238	3426.9311
Test6												
4	59 106 133 153	48 101 149 177	57 107 137 157	1660.3048	6 18 25 227	79 132 163 185	57 107 137 157	1647.4027	59 106 133 153	79 131 163 185	59 109 137 157	1659.9864
8	20 31 44 69 98 121 139 156	41 71 101 129 148 162 176 190	14 31 58 93 119 135 148 162	1707.9832	25 46 68 91 109 125 141 157	40 71 101 128 146 161 174 190	17 35 61 96 120 136 149 163	1710.0436	27 58 87 107 124 137 149 161	40 65 100 128 149 165 179 193	22 50 83 108 127 141 153 166	1712.033
10	4 20 30 52 81 101 118 131 144 158	29 39 40 54 67 71 72 74 140 253	11 24 46 77 101 118 132 144 155 167	1708.8455	10 23 38 59 86 105 120 135 147 160	40 67 95 119 135 148 159 170 181 193	10 21 40 51 74 100 119 133 146 160	1715.7845	21 34 61 89 108 122 134 145 155 165	38 64 93 120 138 152 164 175 186 199	22 42 69 90 116 130 143 155 167 229	1717.9712
12	14 20 22 63 77 79 103 106 196 197 202 236	32 54 79 103 119 127 138 149 161 172 183 194	1 7 19 35 54 74 98 114 128 140 151 163	1714.0984	11 23 38 57 81 98 112 124 134 144 153 162	8 42 46 52 120 127 193 200 216 220 223 227	7 20 40 62 81 99 112 125 136 146 156 167	1717.9691	16 24 36 57 71 92 109 122 134 145 155 165	38 68 94 116 133 146 157 166 176 185 193 202	11 26 45 67 84 104 120 131 140 149 159 170	1722.7707
Leaf1												
4	79 94 113 190	95 112 131 199	52 72 160 247	1821.4726	79 95 114 190	96 116 148 210	48 63 81 187	1825.2078	79 95 114 190	95 112 131 199	48 63 80 161	1827.0865
8	72 84 95 107 127 218 229 250	86 96 105 114 124 137 162 239	44 57 70 87 228 228 233 238	1844.4818	68 78 87 95 104 117 141 204	84 95 104 113 123 136 156 207	36 46 55 65 76 91 99 169	1850.2119	67 77 86 95 106 121 166 213	84 95 104 113 123 136 155 205	40 51 61 73 93 126 130 157	1848.3456
10	41 68 77 85 92 101 112 129 171 198	85 97 108 119 133 161 223 224 228 230	37 47 56 65 75 88 94 97 153 200	1849.6088	67 78 87 96 106 118 146 187 188 196	7 75 94 95 102 111 219 226 227 230	37 47 56 66 78 97 138 181 184 215	1850.4216	68 78 86 94 102 114 129 155 164 192	81 90 98 106 114 123 133 148 169 215	41 52 62 74 91 136 151 197 197 212	1850.8873
12	63 72 79 87 94 102 112 125 159 182 194 209	85 95 104 113 123 135 156 216 235 241 252 253	34 43 50 57 64 72 82 94 117 179 186 190	1852.7437	67 76 84 91 99 108 120 146 226 233 240 243	65 85 95 104 112 122 133 152 195 215 232 244	40 50 59 68 80 101 163 206 234 243 247 247	1851.2717	65 73 80 86 92 98 106 115 128 143 204 251	77 87 95 101 108 115 124 135 152 180 246 247	36 43 51 58 66 75 88 115 133 133 184 186	1854.2527
Leaf2												
4	107 120 131 146	131 143 155 169	98 150 153 173	201.747	21 112 179 196	132 144 155 169	38 140 171 178	201.712	108 120 131 146	34 94 110 166	73 85 97 111	202.712
8	98 106 113 120 127 135 144 157	3 27 84 91 116 135 182 212	23 56 72 112 119 125 165 244	202.1451	45 75 84 101 127 156 204 206	21 61 90 101 128 169 238 245	3 50 93 102 111 126 182 234	202.2451	10 23 74 117 127 194 216 229	29 43 63 101 176 212 216 225	85 92 115 130 133 163 195 196	203.2451
10	9 32 77 104 111 141 151 172 184 234	1 69 88 88 90 100 182 221 239 250	140 141 145 156 164 167 183 200 207 240	204.2147	9 41 71 116 140 148 192 211 219 247	1 24 29 55 136 144 151 184 186 188	12 19 39 42 72 85 160 168 177 233	204.4147	8 54 56 95 99 114 137 168 185 187	20 39 67 80 110 134 140 206 213 228	7 21 35 37 103 134 136 169 174 204	205.4147
12	32 56 71 102 108 134 155 156 183 235 238 243	20 34 77 81 99 146 156 177 197 230 233 253	5 11 45 70 79 91 116 124 124 180 227 238	207.2146	19 27 36 57 59 92 125 137 151 153 157 209	30 53 78 83 84 119 137 151 166 217 225 234	1 4 11 48 75 76 104 113 151 167 237 247	207.5146	6 79 83 104 136 149 168 193 199 206 232 240	16 30 55 98 103 135 147 164 189 206 207 238	24 50 54 57 71 76 77 85 95 206 223 226	208.5146

TABLE 7. The PSNR and FSIM of each algorithm under Otsu.

K	GOA		WOA		FPA		PSO		BA		MGOA	
	PSNR	FSIM	PSNR	FSIM	PSNR	FSIM	PSNR	FSIM	PSNR	FSIM	PSNR	FSIM
Test1												
4	17.8818	0.9261	17.9358	0.9260	17.9488	0.9277	17.9019	0.9261	17.9359	0.9276	17.9296	0.9263
8	19.4774	0.9365	22.8050	0.9425	22.6641	0.9528	22.8599	0.9429	22.9010	0.9628	22.9881	0.9726
10	21.5370	0.9785	24.3637	0.9506	24.5556	0.9606	22.4253	0.8511	25.0574	0.9717	25.1137	0.9801
12	21.0019	0.8953	25.6055	0.9646	25.8297	0.9755	26.5185	0.9659	26.2462	0.9759	26.57137	0.9848
Test2												
4	19.7421	0.8737	22.2953	0.8585	17.6204	0.8249	22.3854	0.8625	9.1549	0.6271	22.3358	0.8596
8	17.5580	0.8203	26.5263	0.9287	28.5775	0.9390	19.4203	0.7473	28.3855	0.9421	28.0452	0.9408
10	28.7655	0.9562	28.5416	0.9444	24.4487	0.8606	24.8518	0.8763	29.9425	0.9556	29.3361	0.9427
12	31.1774	0.9651	29.6880	0.9481	31.6853	0.9645	27.6906	0.9555	22.5105	0.8641	31.1546	0.9661
Test3												
4	20.9637	0.8423	20.9549	0.8435	17.6204	0.8249	21.0447	0.8459	20.9765	0.8440	20.9459	0.8442
8	26.0858	0.9190	25.3875	0.9073	28.5775	0.9390	26.0844	0.9204	26.0932	0.9191	26.0069	0.9183
10	27.9289	0.9395	27.4917	0.9391	24.4487	0.8606	27.5716	0.9330	23.3578	0.9304	27.5634	0.9326
12	28.9142	0.9433	28.8727	0.9456	31.6853	0.9345	25.2230	0.9430	29.2843	0.9464	29.3033	0.9498
Test4												
4	11.9645	0.7359	18.8626	0.8126	15.4385	0.8206	11.2251	0.6075	8.5443	0.5221	18.6919	0.8362
8	20.9723	0.9332	25.9357	0.9440	26.4476	0.9503	22.8623	0.9191	26.2486	0.9576	25.9002	0.9418
10	28.3387	0.9613	26.0914	0.9514	28.1569	0.9573	24.6689	0.8554	23.0649	0.8207	27.7368	0.9584
12	29.7657	0.9764	29.0859	0.9756	21.9930	0.8398	29.3807	0.9761	29.5098	0.9737	29.3753	0.9850
Test5												
4	19.7421	0.8737	18.1592	0.8341	18.1529	0.8345	17.4385	0.8014	18.1753	0.8329	18.1605	0.8338
8	17.5580	0.8203	22.5968	0.8999	20.3895	0.9010	22.6381	0.8985	22.8771	0.9025	22.6538	0.8976
10	28.7655	0.9362	23.6668	0.9129	24.1281	0.9168	19.2965	0.8981	24.0593	0.9170	24.6323	0.9153
12	31.1774	0.9451	24.6542	0.9252	25.6054	0.9368	25.0559	0.9287	26.0815	0.9376	24.9510	0.9567
Test6												
4	14.0959	0.6419	22.2759	0.8815	17.7829	0.8390	22.5257	0.9121	17.1314	0.8829	22.3189	0.8853
8	24.0664	0.9586	27.7305	0.9613	27.6045	0.9547	27.3939	0.9571	27.7441	0.9550	28.0956	0.9621
10	25.5099	0.9658	28.4438	0.9617	29.7000	0.9657	25.1605	0.8695	29.0798	0.9688	29.1909	0.9689
12	25.2030	0.8341	30.0056	0.9724	30.7274	0.9710	25.6596	0.9453	26.4581	0.8576	30.7110	0.9774
Leaf1												
4	17.1566	0.7837	17.2368	0.7811	16.8861	0.7737	17.2368	0.7811	17.1566	0.7837	16.8218	0.7792
8	21.1842	0.8816	21.6482	0.8679	20.6220	0.8509	21.6482	0.8679	21.1842	0.8816	26.9839	0.8828
10	23.7327	0.8870	23.0215	0.9015	22.5213	0.8801	23.0215	0.9015	23.7327	0.8870	25.4394	0.9319
12	27.1349	0.9198	23.0393	0.8986	24.8734	0.9081	23.0393	0.8986	27.1349	0.9198	27.2374	0.9517
Leaf2												
4	14.6327	0.7257	14.2314	0.7594	16.7001	0.7914	14.2314	0.7594	14.6327	0.7257	17.5943	0.8626
8	22.3207	0.9315	22.5914	0.8779	19.6479	0.9246	22.5914	0.8779	22.3207	0.9315	16.8583	0.8827
10	20.9126	0.8007	16.0816	0.8170	23.9895	0.9503	16.0816	0.8170	20.9126	0.8007	21.7665	0.9127
12	25.3912	0.9135	24.3421	0.9232	21.9162	0.8693	24.3421	0.9232	25.3912	0.9435	26.2144	0.9601

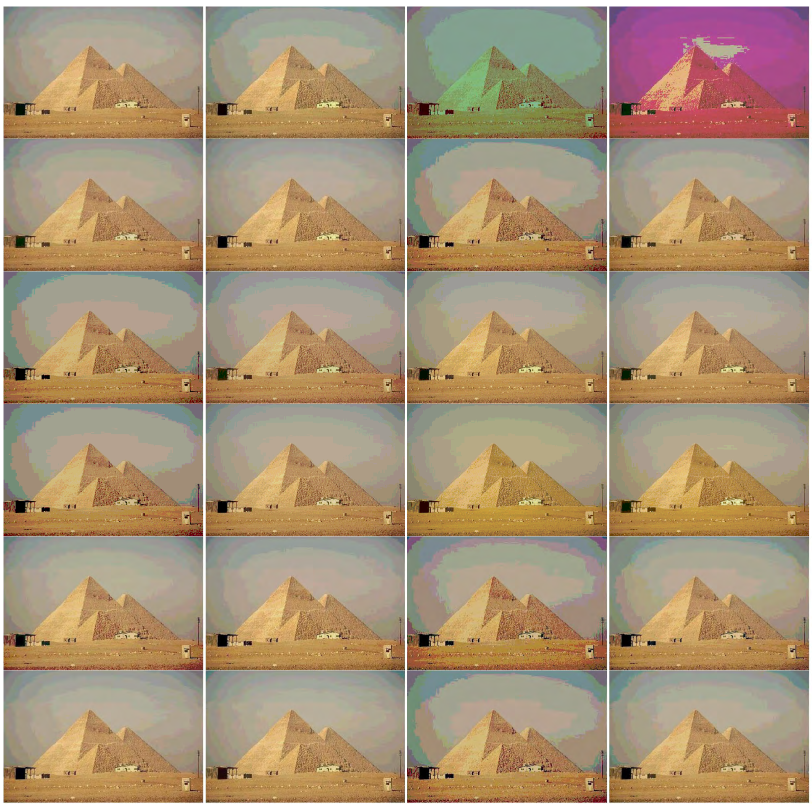

FIGURE 6. The Tsallis results of image test2.

Mousavirad and Ebrahimpour-Komleh [50] proposed a simple but efficient population-based metaheuristic algorithm called Human Mental Search (HMS). The mental search of HMS that explores the region around each solution based on Levy flight. So, Levy flight can increase the global search ability of algorithm and avoid local optimization.

TABLE 8. The optimal fitness value and threshold value of each algorithm under Renyi.

K	GOA				WOA				FPA			
	R	G	B	F	R	G	B	F	R	G	B	F
Test1												
4	40 90 161	46 89 137	48 86 132	1.6126	39 89 161	46 89 136	47 85 130		39 89 161	44 87 134	48 86 132	
	227	208	206		227	208	204	1.4114	227	206	206	1.8794
8				1.8397					21 46 72	16 37 61	23 42 60	
	20 44 68 96	26 46 66 88	29 50 70 91		19 43 69 97	27 50 72 95	28 48 68 89		100 133	85 110	79 103	
	128 175 215	112 140	114 143		131 177	120 147 181	112 139		179 218	138 173	130 168	
	240	176 227	180 227		217 241	227	176 225	1.6558	242	221	223	1.9178
10		24 43 61 79	24 40 57 73	2.043	15 33 53 74		24 40 56 73		12 30 49 69	24 42 60	23 40 56	
	15 33 52 73	98 118 139	91 111 135		96 120 150	22 41 58 75	90 109 130		90 113 141	78 97 117	73 90 109	
	96 120 152	164 192	162 194		186 218	93 112 132	156 190		181 215	138 163	130 156	
	187 219 241	230	232		242	155 183 226	231	2.1013	239	193 232	189 231	2.1683
12		18 33 48 63	17 30 45 61	2.2248	13 30 47 65		21 35 47 60		12 26 42 58	21 37 53	20 33 47	
	1 14 27 42	78 94 112	77 92 109		83 102 126	1 1 22 40 57	75 89 104		75 93 114	67 82 98	62 77 93	
	58 76 92 114	132 150	129 153		153 185	74 93 112	120 133		139 171	115 133	111 130	
	142 179 217	162 192	185 221		209 227	133 158 191	160 193		200 222	150 174	153 182	
	240	230	231		245	233	234	2.3118	243	204 237	212 240	2.3978
Test2												
4	112 161 181	105 141	64 94 120	1.579	45 100 126		64 94 120		115 161	26 102	62 93 119	
	210	162 186	144		127	15 19 62 241	144	1.5311	181 210	214 247	144	1.6312
8		75 110 130		1.869					34 65 90 96	57 89 111	29 43 45	
	35 68 117	145 157	35 62 78 93		42 49 52 58	54 85 105	36 63 81 95		132 134	134 150	54 55 166	
	149 166 180	169 186	109 126		83 146 170	121 139 154	111 127		172 244	165 182	176 190	
	196 219	228	140 152		249	167 188	141 152	1.8214		204		1.8845
10		44 77 102		2.0846					31 47 60 61	65 98 119	24 51 68	
	58 90 131	120 134	37 65 82 96		36 43 85 85	46 76 104	27 58 72 80		81 99 120	134 145	81 93 106	
	150 162 172	147 158	112 127		114 115	122 136 148	91 103 117		125 207	155 164	121 133	
	182 194 210	169 187	138 148		116 206	159 170 187	130 141		246	173 187	144 154	
	229	210	156 171		233 239	227	152	2.1055		212		2.1124
12				2.2909	31 63 77		12 49 96		39 41 44 45	34 54 86	1 33 63 67	
		13 18 43 72	10 34 55 70		109 143		112 117		68 115 135	103 120	82 95 109	
	58 73 74 110	72 108 132	81 93 106		156 165	18 52 73 83	187 203		155 169	134 144	124 137	
	141 156 165	145 157	119 132		174 183	109 128 141	204 204		181 196	153 162	150 187	
	174 183 193	169 186	144 153		193 210	152 162 172	223 233		219	171 186	190	
	210 229	209	176		231	186 210	237	2.3154		213		2.3296
Test3												
4	48 86 137	43 77 109	73 120 156	1.7055	40 70 105	6 141 169	44 55 220		115 161	26 102	62 93 119	
	256	148	187		151	182	249	1.6501	181 210	214 247	144	1.716
8			40 61 87	1.8714			75 83 94		34 65 90 96	57 89 111	29 43 45	
	1 30 49 67	23 42 61 77	117 141		27 47 63 80	20 39 59 76	205 207		132 134	134 150	54 55 166	
	86 108 132	93 113 135	161 182		99 119 143	93 113 135	210 221		172 244	165 182	176 190	
	166	163	206		173	163	248	1.8214		204		1.9707
10			42 63 84	2.1288					31 47 60 61	65 98 119	24 51 68	
		22 41 58 71	107 129		23 40 53 67		35 50 69 93		81 99 120	134 145	81 93 106	
	19 35 50 64	83 97 111	147 162		83 99 116	6 20 39 56	116 133		125 207	155 164	121 133	
	81 100 118	127 145	177 194		135 159	68 79 93 112	150 166		246	173 187	144 154	
	140 167 192	171	216		187	134 163	185 209	2.1221		212		2.2042
12				2.3831					39 41 44 45	34 54 86	1 33 63 67	
		21 37 54 68	35 54 76 98		19 33 46 57		33 47 58 72		68 115 135	103 120	82 95 109	
	15 30 44 56	81 94 110	117 130		69 84 99	21 38 55 66	86 100 121		155 169	134 144	124 137	
	70 85 102	126 144	144 158		115 133	78 91 106	138 156		181 196	153 162	150 187	
	118 136 155	168 218	171 185		150 172	121 134 153	171 189		219	171 186	190	
	175 194	218	200 217		195	178 228	212	2.2145		213		2.4244
Test4												
4	103 136 167	92 125 155	75 105 131	1.6661	103 136	93 126 157	75 105 131		101 133	2 32 39	4 38 57	
	195	184	157		167 195	185	157	1.5775	165 194	203	210	1.6743
8		47 77 99	53 75 92	1.9035	66 92 110		52 75 93		55 82 105	59 81 98	47 71 90	
	37 82 104	119 140	109 124		128 147	70 92 111	112 129		126 146	119 139	106 121	
	124 145 166	160 179	137 152		167 187	130 148 165	145 166		167 187	159 177	135 149	
	186 206	197	170		206	180 198	256	1.8521	206	197	168	1.9235
10		39 57 81		2.0507	69 91 107				60 81 100	29 57 76	9 16 29 31	
		101 118	1 1 52 75		123 139		1 9 53 75		118 135	92 107	159 165	
	29 69 93 109	138 157	93 109 124		155 170	60 81 98 113	92 108 122		152 168	125 143	169 191	
	127 145 160	174 194	138 153		183 196	129 144 158	136 151		182 196	159 176	192 192	
	177 192 209	235	171		211	171 183 199	170	2.0135	211	195		2.1835
12				2.2355					37 70 88	35 36 59	1 30 45 50	
		3 58 76 92	1 42 56 70		31 34 57 58		30 51 68 83		101 111	77 93 106	68 85 100	
	50 74 90 105	106 121	83 98 111		84 105 124	1 1 1 40 67	97 110 122		123 138	121 138	113 126	
	118 131 146	137 151	124 137		143 162	87 104 120	133 144		154 169	152 167	139 154	
	160 174 187	164 177	152 170		178 193	139 158 176	156 171		182 196	181 197	172	
	199 213	190 205	218		210	195	187	2.2307	212			2.4107
Test5												
4	51 91 147	51 83 136	51 81 132	1.5512	50 89 145	50 81 132	51 81 132		50 90 145	50 82 134	51 81 131	
	213	206	205		211	204	205	1.5885	212	205	205	1.6775
8				1.8077					34 48 65 89	42 55 71	41 53 66	
	35 50 70 94	41 55 71 92	41 53 66 84		34 49 68 92	40 53 68 88	40 52 65 83		116 149	92 118	84 110	
	120 153 190	118 150	108 141		118 151	113 144 182	109 142		187 229	149 185	143 181	
	230	187 229	177 223		189 230	225	180 225	1.6779		226	225	1.9339
10		39 50 61 76	38 49 59 72	2.0112	32 42 56 77		36 46 56 66		31 41 53 70	38 49 60	37 47 57	
	31 42 53 69	95 119 145	90 113 122		101 125	31 34 80 85	82 104 130		93 118 147	75 93 115	69 86 110	
	88 110 132	175 204	152 189		149 172	91 94 101	156 185		176 204	142 173	138 169	
	163 199 233	235	227		197 229	208 229 255	225	2.0214	233	203 233	202 229	2.1591
12		38 48 59 74	36 45 54 63	2.2987	30 39 51 64		37 46 55 65		30 38 49 61	37 46 55	35 44 52	
	31 41 53 70	94 115 139	76 93 115		79 97 100	37 46 55 65	77 94 113		76 95 117	66 79 96	60 68 80	
	91 110 126	147 167	139 164		119 141	79 96 114	134 148		139 163	117 141	95 115	
	149 181 196	184 206	186 209		167 199	137 162 186	174 202		180 205	148 173	141 172	
	221 235	235	238		235	211 239	233	2.2114	234	202 232	202 233	2.3896
Test6												
4	59 106 133	80 133 164	58 109 137	1.6778	54 99 128	48 101 149	58 109 137		58 104 131	48 100	57 108	
	153	186	157		149	177	157	1.6124	151	149 176	137 157	1.6853
8		41 68 97	26 57 95	1.8645			19 41 71		28 44 49 59	38 63 88	19 39 67	
	27 56 86 108	125 145	123 141		13 26 49 73	39 65 97 130	102 123		74 87 203	115 137	95 116	
	124 138 150	160 176	158 181		97 119 138	152 170 188	138 151		232	155 171	131 145	
	162	191	182		154	254	164	1.9012		188	161	1.9355
10				2.1354					16 31 55 76	29 48 72	12 26 49	
		27 47 70 99	18 37 61 86		16 31 58 86		19 41 70 97		95 109 122	98 126	78 103	
	18 31 53 75	125 142	107 122		106 122	38 62 91 119	114 125		133 145	146 162	119 131	
	96 113 127	156 169	133 143		136 148	138 151 162	135 145		158	176 191	141 152	
	139 151 162	181 194	154 167		160 206	174 185 196	155 166	2.1222		245	165	2.1851
12		30 44 53 64	20 39 62 85	2.3198	12 23 36 56		16 31 47 73		4 26 51 75	8 22 30	19 20 39	
	22 44 71 93	86 108 126	102 116		79 98 112	31 48 65 81	96 111 125		96 111 125	43 57 74	56 67 92	
	108 122 133	143 157	127 136		125 136	104 124 139	136 146		137 149	78 113	113 129	
	143 154 165	170 182	144 152		146 155	152 164 175	156 167		161 208	139 157	141 152	
	208 229	194	160 170		165	186 196	215	2.3457	215	174 190	165 222	2.4125

TABLE 8. ***(Continued.)*** **The optimal fitness value and threshold value of each algorithm under Renyi.**

Leaf1												
4	77 93 112 189	95 111 130 198	47 62 80 172	1.4112	78 94 112 189	94 111 129 197	47 62 80 172	1.5278	78 94 113 188	95 111 130 198	47 62 80 172	1.4775
8	67 77 86 95 104 116 138 240	84 95 104 113 124 141 176 216	26 34 165 168 183 190 222 242	1.8077	49 66 67 196 203 213 213 215	81 92 102 112 122 136 164 216	37 47 57 67 79 103 189 245	1.6645	65 76 85 94 103 117 146 196	83 93 103 112 122 135 158 233	21 29 122 161 183 199 220 247	1.6339
10	63 72 80 87 94 102 112 126 147 209	78 87 94 101 109 116 124 135 156 241	39 48 57 66 76 89 118 167 195 238	2.0512	65 74 82 90 98 107 120 141 197 243	77 86 94 102 109 118 128 141 168 205	26 122 131 155 166 180 218 225 251 254	2.1014	63 72 79 86 94 102 112 127 155 210	78 89 97 106 115 125 139 171 237 249	24 31 97 100 131 177 210 233 241 241	2.0191
12	57 68 77 84 92 100 109 124 148 173 233 247	78 86 94 102 109 117 126 138 159 204 233 248	36 45 53 61 70 82 100 131 185 211 235 247	2.1577	59 66 76 84 91 99 109 122 148 225 247 248	78 87 94 100 107 113 121 130 143 164 231 245	35 43 51 58 65 74 85 102 137 200 209 246	2.3111	61 70 78 86 94 102 112 129 161 202 220 237	61 84 84 132 138 148 187 195 199 221 230 255	34 43 51 59 67 76 88 107 166 224 240 247	2.1596
Leaf2												
4	107 119 130 145	42 68 147 169	74 86 98 112	1.6118	108 120 131 146	59 68 78 223	74 86 97 111	1.3012	108 120 131 146	109 135 149 165	81 184 209 209	1.5453
8	42 46 49 80 171 186 186 250	15 79 151 155 176 188 210 253	28 70 77 112 153 167 175 231	1.8135	5 12 48 57 127 178 189 206	2 37 49 65 76 191 230 248	2 10 41 138 178 193 219 237	1.5077	10 120 132 133 137 166 234 241	10 103 161 163 178 212 228 244	85 113 131 133 190 197 204 245	1.6455
10	17 27 31 39 43 65 109 224 241 246	7 21 25 46 52 62 76 99 213 243	1 1 34 43 47 104 135 161 190 247	2.1474	15 26 31 51 64 108 122 127 133 245	3 25 52 64 75 77 142 151 172 180	36 130 132 166 185 211 218 223 229 245	1.8512	6 19 51 75 85 131 160 176 218 230	15 20 29 45 66 83 105 124 189 216	6 12 16 23 28 30 51 106 160 173	2.0251
12	46 55 115 140 145 189 208 216 217 219 241 248	8 19 39 88 94 95 117 199 201 206 239 246	4 24 53 94 114 138 187 194 195 206 242 243	2.3018	25 50 51 68 72 81 106 110 129 146 157 202	41 64 65 84 95 125 149 172 195 200 242 255	10 14 34 39 45 83 90 157 165 225 241 248	2.0177	39 61 68 73 78 96 125 128 143 152 158 163	22 45 62 76 79 140 167 173 200 203 210 212	9 47 50 71 91 116 137 152 212 212 223 229	2.2225

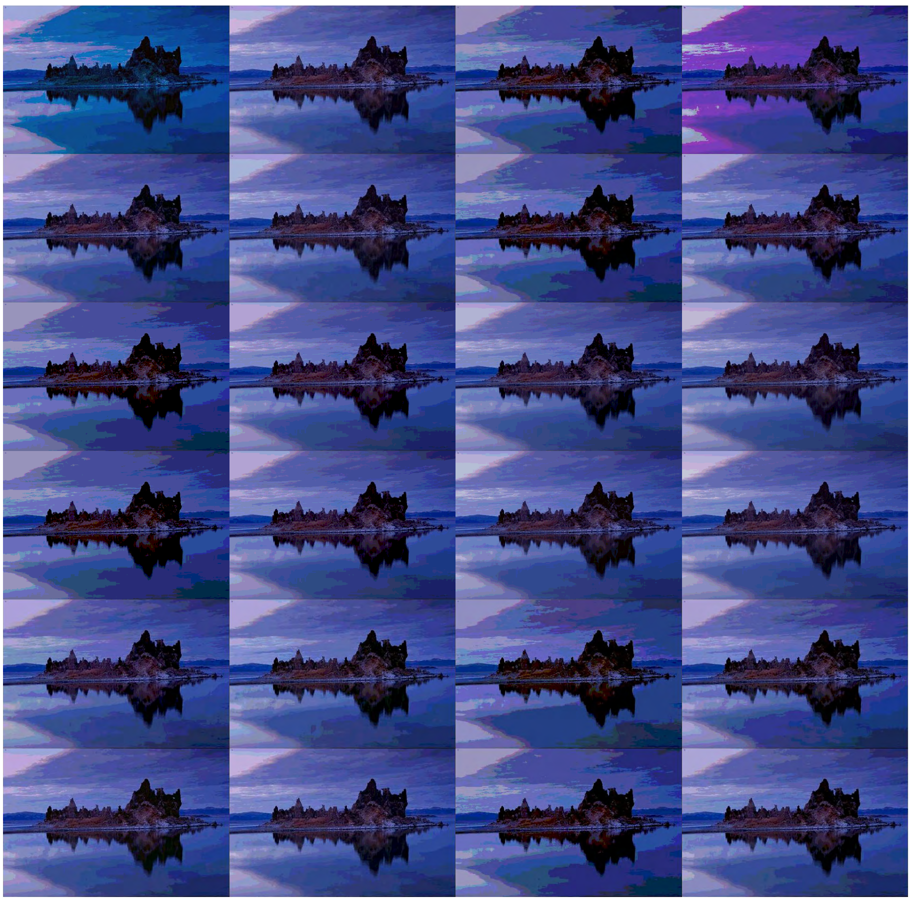

FIGURE 7. **The Tsallis results of image test3.**

In order to balance the exploration and exploitation of GOA, it's a better solution to use Levy flight to improve GOA algorithm.

In this paper, a modified GOA (MGOA) is proposed. The main contribution of this study is that the Levy flight algorithm improves the original GOA for multilevel threading.

TABLE 9. The optimal fitness value and threshold value of each algorithm under Renyi.

K	PSO				BA				MGOA			
	R	G	B	F	R	G	B	F	R	G	B	F
Test1												
4	38 44 58	38 49	47 85	1.6768	59 105	80 133	59 109 137	1.6911	40 90 161	46 89	48 86	1.7911
	98	57 89	131 204		133 153	164 186	157		227	137 208	132 206	
8	17 37 58	28 51	27 47 67	1.9416	27 51 79	9 13 43	24 56 91	1.9175		27 50	27 46 65	2.0175
	80 107	73 96	88 111		103 122	90 130	115 131		20 45 71 98	71 93	86 109	
	140 184	121 149	140 178		136 148	155 173	143 155		130 177	117 144	135 170	
	231	183 227	226		161	190	167		217 241	179 227	222	
10	13 29 49	23 41	23 39 56	2.1672	25 34 56	22 34 47		2.1678		23 41	24 40 57	2.2678
	69 90	59 76	72 89		84 103	80 114	18 38 66		14 31 49 66	59 77	73 89	
	114 144	95 114	108 129		120 132	136 153	82 106 123		87 112 144	96 116	108 131	
	184 219	135 159	154 187		143 153	168 181	136 146		182 215	138 164	160 195	
	241	188 228	230		163	193	157 169		239	196 234	235	
12	15 33 53	21 38		2.3479	22 39 61	23 35 45		2.3964		17 30		2.4764
	72 91	53 69	21 35 49		84 100	57 72	6 18 20 24		13 31 48 66	42 57	21 35 49	
	113 140	84 100	63 76 90		113 124	101 128	39 64 94		85 105 125	71 84	63 77 93	
	176 204	116 135	107 125		134 143	146 160	116 131		151 184	100 119	110 129	
	217 231	154 177	146 171		151 159	173 184	144 155		209 227	139 163	152 181	
	246	206 237	201 237		170	196	168		245	193 232	214 239	
Test2												
4	110 159	70 84	59 90	1.6334	112 161	92 137	56 90 116	1.6281	115 161	103 140	63 94	1.7681
	180 208	246 249	117 143		181 211	160 185	144		181 210	161 185	120 144	
8	62 100	62 105	19 52 73	1.8844	95 140	52 103		1.8836	74 116 145	40 46	1 40 69	1.9136
	142 159	126 141	88 103		158 170	129 143	1 6 54 80		160 172	90 123	85 100	
	171 183	154 167	121 137		182 194	157 169	98 117 136		184 199	141 156	119 136	
	198 221	183 206	150		211 232	185 203	150		222	169 188	150	
10	71 95	49 77	29 56 72	2.1125	34 37 90	71 108		2.109		43 48	36 62 79	2.219
	129 149	107 126	84 94		143 159	128 141	4 18 53 74		31 39 60 94	78 99	91 104	
	162 172	139 149	106 119		171 183	151 162	87 103 118		144 159	121 136	118 132	
	182 193	159 170	133 144		195 211	172 186	131 143		171 184	149 161	144 153	
	208 228	184 204	153		231	205 214	154		201 224	172 191	176	
12	88 111	22 50		2.3288	31 49 81	41 69		2.3268	31 42 78	12 12	9 45 68	2.4168
	133 145	89 112	1 1 25		93 127	100 119	3 17 32 43		108 124	21 42	82 94	
	156 164	127 138	28 61 80		148 155	133 143	57 80 91		145 159	76 97	105 118	
	173 182	148 157	94 112		166 177	152 162	170 197		171 181	118 133	129 138	
	190 200	165 172	127 140		190 209	171 184	229 237		193 210	147 158	147 155	
	215 234	185 210	151 193		233	198 225	245		231	170 188	174	
Test3												
4	40 70	41 74	72 119	1.7148	40 71 106	43 77	63 110 151	1.7161	40 71 106	43 76	72 119	1.6961
	105 151	105 144	155 187		152	109 148	186		152	108 147	155 187	
8	27 45 62	24 46	39 59 87	1.9707		21 38 61	49 68 101	1.9541		31 55	52 86	2.1541
	81 103	67 85	117 137		1 29 49 66	77 94	130 152		29 49 67 86	73 90	117 141	
	128 163	106 130	156 177		86 108	114 136	169 188		107 133	108 128	160 180	
	235	160 217	203		134 168	164	213		168 256	148 173	203 237	
10		19 33	35 55 76	2.2045				2.2048		21 40	40 62 71	2.3148
	20 35 49	47 61	100 124		24 39 53	1 23 42	5 37 64 94		24 40 53 67	58 72	93 119	
	64 80 97	76 91	143 160		68 86 104	59 75 90	122 143		83 100 118	85 100	140 157	
	114 133	108 127	176 193		122 138	107 124	160 177		138 161	117 135	172 188	
	157 184	146 170	215		161 186	145 171	194 217		186	155 177	211	
12		19 32	34 52 67	2.4249				2.4251		18 31	34 51 65	2.5051
	18 31 44	47 61	83 101		18 31 44	1 23 40	4 21 39 59		1 24 39 50	43 58	80 99	
	54 65 76	73 84	122 140		56 69 85	57 70 82	80 105 127		61 73 87	70 80	117 134	
	88 101	96 109	155 169		101 118	95 110	145 160		103 119	92 105	149 162	
	117 132	123 139	184 199		137 157	127 145	176 193		138 163	119 135	176 193	
	153 181	155 177	216		179 201	169 213	214		188	153 175	216	
Test4												
4	102 135	12 38	75 105	1.6743	103 136	94 127	76 106 131	1.6743	103 136	93 126	75 105	1.6943
	166 195	239 252	131 157		167 195	157 185	157		167 195	157 185	131 157	
8	53 80 97	57 77	51 71 88	1.9236	35 59 98	70 91	51 72 91	1.9341	12 71 99	53 82	1 57 80	1.9841
	114 136	94 114	107 125		122 145	111 130	108 123		120 142	101 121	99 117	
	157 180	135 154	142 163		167 187	149 165	137 153		165 186	140 160	133 150	
	202	174 195	236		206	181 199	172		206	178 198	170	
10	66 89	58 76	31 54 72	2.1516	67 88 106	35 64 88		2.1837		36 66	47 68 83	2.2237
	107 123	91 107	88 103		123 139	107 126	24 43 68		14 45 70 85	87 105	98 113	
	139 156	123 139	117 131		154 171	143 159	86 103 117		104 125	122 138	126 139	
	171 185	155 169	143 158		185 199	173 186	130 142		145 166	154 169	152 170	
	198 213	183 199	175		214	203	156 174		184 205	183 199	196	
12	33 37 57	18 23		2.3813				2.4079		30 56	11 53 75	2.4/79
	83 102	31 64	28 46 63		11 54 64	26 40 55	31 52 70		11 57 72 91	76 92	94 102	
	118 134	87 106	76 88 99		83 96 110	70 87 99	81 94 107		105 120	103 115	122 131	
	149 165	125 143	112 124		121 138	113 129	119 130		137 153	130 145	156 163	
	179 193	162 179	136 149		156 175	147 165	141 154		169 184	160 174	181 197	
	209	198 245	164 180		191 208	180 198	172 227		197 214	187 202	214	
Test5												
4	50 90	50 82	51 80	1.6772	51 91 148	51 83	51 81 132	1.6775	51 91 148	51 83	51 81	1.6975
	147 213	135 206	131 204		214	136 206	205		214	135 205	131 204	
8	34 49 68	41 53	41 53 66	1.9339	35 50 69	41 55 71		1.9341		42 55	42 54 67	1.9841
	91 117	66 85	83 107		93 119	93 120	41 54 69		34 49 69 93	70 91	85 110	
	149 186	110 142	137 176		153 193	151 186	89 113 144		118 148	116 146	143 181	
	226	180 225	224		232	228	182 227		183 227	181 225	225	
10	32 42 54	39 49	39 50 61	2.1665		39 50 61		2.1603		39 49	40 51 62	2.2203
	72 93	61 76	75 93		32 42 55	76 95	9 41 52 63		32 42 54 72	60 77	75 92	
	114 136	95 118	117 142		72 93 115	116 141	78 99 122		95 119 148	97 120	113 139	
	165 198	143 172	168 193		140 171	170 200	151 186		182 202	143 172	167 192	
	232	202 235	223		204 236	234	228		233	199 231	227	
12	31 41 52	39 49		2.3898				2.3932		37 46		2.4232
	68 85	59 74	36 44 52		30 39 49	36 45 53	35 43 53		31 41 53 71	55 65	36 45 54	
	107 125	91 113	60 69 81		62 78 97	62 76 94	63 76 92		89 112 136	81 99	64 78 94	
	148 157	138 163	97 116		115 135	115 140	108 133		163 189	119 142	115 142	
	180 207	179 202	141 169		159 183	165 186	162 180		208 217	162 176	162 178	
	235	218 238	202 234		210 237	196 230	212 228		238	199 231	204 230	
Test6												
4	59 105	21 190	55 105	1.6853	59 106	81 134	54 103 137	1.6852	59 105 132	48 101	58 109	1.7052
	132 152	244 252	136 157		133 153	164 186	157		152	149 177	137 157	
8	27 55 83	25 48	16 37 65	1.9346	18 50 81	42 72	22 50 83	1.9351	28 69 108	30 52	18 33 59	1.9851
	104 122	80 114	95 115		105 123	109 135	110 127		134 153	87 126	91 117	
	136 149	140 158	130 143		137 149	153 166	141 153		219 219	152 171	134 148	
	161	174 189	158		161	180 193	165		232	189 243	163	
10		35 55	21 39 60	2.1699	23 48 72	11 38 70		2.1847		25 32	20 41 68	2.2247
	3 6 26	72 93	75 96		91 109	101 129	18 44 76		20 39 63 90	38 56	92 109	
	44 64 91	115 133	112 125		123 135	148 161	99 116 128		110 125	80 109	123 135	
	111 126	149 162	138 150		145 155	173 184	139 149		139 152	135 155	145 156	
	141 156	176 191	163		166	196	158 169		164 169	173 190	168	
12		6 7 32	16 32 49	2.3949		24 31 50		2.4191		25 43		2.4591
	16 28 31	34 35	71 89		8 24 39 50	68 94	5 13 22 42		20 26 45 67	59 81	1 16 28	
	41 44 45	55 84	105 120		68 88 105	125 143	66 93 107		87 105 119	99 125	46 73 98	
	70 96	118 142	131 140		120 132	156 169	122 134		131 142	142 155	116 130	
	115 130	158 174	149 158		143 154	182 194	144 155		152 162	166 176	140 149	
	144 158	190	168		164	228	167		213	187 198	158 169	

TABLE 9. *(Continued.)* **The optimal fitness value and threshold value of each algorithm under Renyi.**

Leaf1												
4	78 94 113 188	95 111 130 198	47 62 80 172	1.5768	78 94 113 188	95 111 130 198	47 62 80 172	1.4775	78 94 112 189	206 222 226 237	47 62 80 165	1.6675
8	65 76 85 94 103 117 146 196	83 93 103 112 122 135 158 233	21 29 122 161 183 199 220 247	1.7416	65 76 85 94 103 117 146 196	83 93 103 112 122 135 158 233	21 29 122 161 183 199 220 247	1.6741	70 80 89 98 109 125 200 253	83 94 103 113 123 136 161 208	35 43 50 58 67 79 99 215	1.8941
10	63 72 79 86 94 102 112 127 155 210	78 89 97 106 115 125 139 171 237 249	24 31 97 100 131 177 210 233 241 241	2.1152	63 72 79 86 94 102 112 127 155 210	78 89 97 106 115 125 139 171 237 249	24 31 97 100 131 177 210 233 241 241	2.1003	42 42 46 55 152 168 206 223 237 253	12 21 51 89 97 167 174 207 211 224	33 40 49 58 67 78 96 158 211 248	2.2103
12	61 70 78 86 94 102 112 129 161 202 220 237	61 84 84 132 138 148 187 195 199 221 230 255	34 43 51 59 67 76 88 107 166 224 240 247	2.3029	61 70 78 86 94 102 112 129 161 202 220 237	61 84 84 132 138 148 187 195 199 221 230 255	34 43 51 59 67 76 88 107 166 224 240 247	2.3112	68 78 86 95 104 116 136 214 228 237 246 247	78 87 95 102 109 116 124 136 153 188 216 238	36 43 50 56 63 71 81 97 134 184 218 242	2.4212
Leaf2												
4	107 119 130 144	142 171 227 240	74 86 98 112	1.6184	108 120 132 146	132 145 156 170	74 86 97 112	1.6222	42 48 138 156	131 143 154 168	72 83 94 109	1.6822
8	71 78 83 89 149 178 201 254	46 47 80 84 126 162 186 206	36 37 39 97 154 177 216 222	1.8454	54 85 88 91 93 96 153 190	26 126 148 195 204 205 230 248	47 48 112 118 153 156 159 244	1.9151	6 22 29 54 57 70 92 190	120 133 146 155 180 229 249 249	16 30 32 34 70 151 225 239	1.9551
10	4 11 71 73 103 153 174 179 196 245	7 8 52 93 123 186 207 211 212 217	42 48 65 81 134 214 230 231 235 243	2.1325	25 44 62 70 83 89 134 140 146 157	17 27 34 36 59 88 177 180 198 227	21 32 68 136 155 184 185 227 232 245	2.1197	14 15 146 190 194 196 221 229 234 241	62 66 86 95 172 186 225 230 246 252	25 27 48 56 67 90 121 182 189 196	2.2297
12	46 55 115 140 145 189 208 216 217 219 241 248	8 19 39 88 94 95 117 199 201 206 239 246	4 24 53 94 114 138 187 194 195 206 242 243	2.3188	17 33 48 96 135 163 178 181 202 205 232 251	9 11 38 51 95 97 113 149 155 161 227 244	19 27 66 143 163 186 192 210 212 237 239 244	2.4321	30 35 35 49 49 51 78 93 179 184 214 230	6 13 15 41 62 69 96 107 157 167 243 251	25 57 62 84 85 91 149 157 177 188 194 226	2.4621

TABLE 10. **The PSNR and FSIM of each algorithm under Renyi.**

K	GOA		WOA		FPA		PSO		BA		MGOA	
	PSNR	FSIM	PSNR	FSIM	PSNR	FSIM	PSNR	FSIM	PSNR	FSIM	PSNR	FSIM
Test1												
4	17.9296	0.9263	17.8681	0.9258	17.8662	0.9285	12.4376	0.7830	21.3080	0.8824	22.3080	0.9263
8	23.0286	0.9724	22.8878	0.9731	22.4109	0.9745	22.5914	0.9717	23.4871	0.9545	27.4871	0.9728
10	24.6216	0.9802	24.2363	0.9805	24.6102	0.9803	24.2492	0.9806	25.4688	0.9690	29.4688	0.9803
12	24.8126	0.9837	25.8030	0.9825	26.8544	0.9851	26.5519	0.9848	29.1940	0.9767	30.1940	0.9856
Test2												
4	22.2771	0.8595	15.1277	0.7285	19.3802	0.8344	18.6569	0.8244	22.4169	0.8636	22.4557	0.8617
8	28.2307	0.9376	26.0257	0.9237	19.4067	0.9062	28.1927	0.9402	27.4345	0.9346	28.0106	0.9297
10	30.0147	0.9556	24.7957	0.9341	25.4089	0.9432	30.3121	0.9539	28.8139	0.9447	29.0423	0.9431
12	30.1630	0.9474	26.3972	0.9553	29.9220	0.9576	30.5325	0.9652	25.3948	0.9625	30.5465	0.9665
Test3												
4	20.2163	0.8364	13.5796	0.7831	19.3802	0.8344	20.9781	0.8478	21.0444	0.8466	20.9653	0.8453
8	25.8759	0.9163	21.8535	0.9117	19.4067	0.9062	25.4781	0.9100	25.6467	0.9130	25.2361	0.9120
10	27.7244	0.9383	27.5833	0.9339	25.4089	0.9432	27.8891	0.9385	27.3997	0.9322	27.5955	0.9362
12	28.7539	0.9429	28.7600	0.9449	29.9220	0.9576	29.2280	0.9472	28.6642	0.9452	29.9908	0.9572
Test4												
4	18.9210	0.8111	18.8626	0.8126	12.1706	0.7138	15.0806	0.7903	18.7588	0.8121	18.8626	0.8126
8	26.4275	0.9559	25.7093	0.9479	26.6283	0.9520	26.0831	0.9570	25.7815	0.9436	25.7962	0.9539
10	27.3394	0.9605	27.4965	0.9602	21.8566	0.9568	28.2332	0.9680	27.9805	0.9681	27.8507	0.9669
12	28.1417	0.9614	28.5762	0.9689	27.2987	0.9727	28.4225	0.9566	29.0029	0.9669	29.2621	0.9737
Test5												
4	18.1562	0.8340	18.2268	0.8323	17.1706	0.7799	18.2417	0.8317	18.1605	0.8338	18.1525	0.8343
8	22.6193	0.9012	22.8777	0.9032	20.8633	0.9021	22.6203	0.8992	22.6568	0.9010	22.4891	0.8980
10	24.0284	0.9150	23.0658	0.8816	23.0781	0.9417	23.8091	0.9133	24.2102	0.9144	23.7118	0.9129
12	24.8455	0.9242	24.8944	0.9283	24.3409	0.9196	24.7497	0.9222	25.2018	0.9316	25.8698	0.9573
Test6												
4	22.2933	0.8833	22.6628	0.9094	19.4267	0.8477	17.4384	0.6777	22.1734	0.8836	22.5840	0.9113
8	27.3076	0.9600	27.1678	0.9471	14.2931	0.8033	27.9686	0.9576	27.9945	0.9591	26.0982	0.9458
10	29.8048	0.9715	29.4423	0.9695	18.7549	0.8879	28.8938	0.9628	29.6450	0.9731	28.9801	0.9664
12	30.5785	0.9734	30.4966	0.9782	23.4181	0.8995	29.3505	0.9626	30.1749	0.9745	30.8499	0.9850
Leaf1												
4	17.2976	0.7822	17.2975	0.7852	18.2401	0.8335	17.1706	0.7799	17.1706	0.7799	14.3835	0.8684
8	21.4036	0.8893	22.9775	0.9016	22.5990	0.8974	20.8633	0.9021	20.8633	0.9021	23.1511	0.8898
10	25.5831	0.9368	22.2286	0.9385	24.2598	0.9170	23.0781	0.9417	23.0781	0.9417	22.7380	0.8107
12	28.9057	0.9535	28.7008	0.9503	25.1718	0.9250	26.3409	0.9196	26.3409	0.9196	29.1205	0.9680
Leaf2												
4	16.7051	0.7750	15.3362	0.8313	22.5728	0.9101	15.2907	0.6658	16.7110	0.7375	22.5728	0.9101
8	17.5233	0.8336	13.3846	0.8293	22.9111	0.9208	18.5841	0.8819	20.0617	0.8415	22.9111	0.9208
10	17.0585	0.8123	18.3913	0.8392	26.3657	0.9432	21.2289	0.7774	14.5217	0.7963	29.3657	0.9632
12	25.3912	0.9635	24.8362	0.9585	28.0390	0.9596	19.0325	0.9241	22.1719	0.9039	29.0390	0.9696

Experiments were performed on natural images and plant stomata images. The proposed algorithm was compared with the original GOA and other five state-of-the-art metaheuristic methods. This paper used two indicators, namely the Peak signal to noise ratio (PSNR) and Feature similarity index for the image (FSIM), to evaluate the performance of the proposed method. The results showed the superiority of the proposed algorithm in terms of the objective function value, image quality measures on both normal and high-level thresholding.

This paper firstly introduces three image segmentation methods: Tsallis entropy, Otsu, and Renyi's entropy. At the same time, three kinds of single threshold segmentation methods are promoted to multi-threshold segmentation method. Then, the modified GOA algorithm is introduced. Finally, we compare with other optimization algorithms to optimize the multi-threshold image segmentation method. In this paper, the applicability of MGOA segmentation method is analyzed and compared, and it is concluded that MGOA is the best in image segmentation.

TABLE 11. P-value of Wilcoxon test comparative Tsallis based method.

Images	K	Wilcoxon test				
		MGOA VS GOA	MGOA VS WOA	MGOA VS FPA	MGOA VS PSO	MGOA VS BA
Test1	4	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
	8	P>0.05	P<0.05	P<0.05	P<0.05	P<0.05
	10	P<0.05	P>0.05	P<0.05	P<0.05	P>0.05
	12	P<0.05	P<0.05	P>0.05	P<0.05	P<0.05
Test2	4	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
	8	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
	10	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
	12	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Test3	4	P<0.05	P>0.05	P<0.05	P<0.05	P<0.05
	8	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
	10	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
	12	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Test4	4	P>0.05	P<0.05	P<0.05	P<0.05	P<0.05
	8	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
	10	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
	12	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Test5	4	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
	8	P<0.05	P<0.05	P>0.05	P<0.05	P<0.05
	10	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
	12	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Test6	4	P>0.05	P<0.05	P<0.05	P<0.05	P<0.05
	8	P<0.05	P>0.05	P<0.05	P<0.05	P<0.05
	10	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
	12	P<0.05	P<0.05	P<0.05	P<0.05	P>0.05
Leaf1	4	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
	8	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
	10	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
	12	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Leaf2	4	P>0.05	P<0.05	P<0.05	P<0.05	P<0.05
	8	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
	10	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
	12	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05

II. PROBLEM ASSESSMENT OF MULTILEVEL THRESHOLDING

The process of searching optimal thresholding values of a given image is considered as a constrained optimization problem [51]. For bi-level thresholding, the problem is to find an optimal value T^*. If the image intensity $I_{i,j}$ is less than the value T^*, the pixel in an image is replaced with a black pixel or a white pixel if the image intensity is greater than that constant T^*, the expression can be stated as follows:

$$g(x,y)=\begin{cases}1 & if\ \ f(x,y)>T^*\\ 0 & if\ \ f(x,y)<T^*\end{cases}\tag{1}$$

The problem can be extended to multilevel thresholding that has more than one threshold value and divide the original image into multiple classes (2):

$$\begin{aligned}N_0&=\{g(x,y)\in I\ |0\le g(x,y)\le t_1-1|\}\\ N_1&=\{g(x,y)\in I\ |t_1\le g(x,y)\le t_2-1|\}\\ N_i&=\{g(x,y)\in I\ |t_i\le g(x,y)\le t_i-1|\}\\ N_n&=\{g(x,y)\in I\ |t_n\le g(x,y)\le L-1|\}\end{aligned}\tag{2}$$

where N_i is the ith class, n is the number of threshold values, and $t_i(i=1,\cdots,n)$ is the ith threshold value.

A. TSALLIS ENTROPY METHOD

Assume that the size of image f is M $\times$ N, its gray levels are $0,1,\ldots L-1$, their probability distributions are $p_0,p_1,\ldots,p_{L\text{-}1}$, and $\sum_{i=1}^{L\text{-}1}p_i=1$. The gray level threshold t divide pixels of image f into object category $C_o=\{(m,n)|f(m,n)=0,1,\cdots,t\}$ and background category $C_B=\{(m,n)|f(m,n)=t+1,t+2,\cdots,L-1\}$. The Tsallis prior entropy of object and background $S_q^O(t)$ and $S_q^B(t)$ are defined as below.

$$S_q^O(t)=\frac{1-\sum_{i=0}^{t}(\frac{p_i}{P_t})^q}{q-1},\quad S_q^B(t)=\frac{1-\sum_{i=t+1}^{L-1}(\frac{p_i}{1-P_t})^q}{q-1}\tag{3}$$

where a parameter q is used to describe nonextensitivity and its optimal value is 0.8 in the threshold section. The total Tsallis entropy of object and background $S_q(t)$ is:

$$S_q(t)=S_q^O(t)+S_q^B(t)+(1-q)S_q^O(t)S_q^B(t)\tag{4}$$

When the Tsallis entropy $S_q(t)$ reaches the maximum, the corresponding gray level t^* is regarded as the optimal threshold.

$$t^*=\underset{0\le t\le L-1}{Arg\max}\{S_q(t)\}\tag{5}$$

The above single threshold selection based on Tsallis entropy can be extended to the situation of multilevel

FIGURE 8. **The Tsallis results of image test4.**

thresholds selection. The n-1 gray level threshold $t_1, t_2, \cdots, t_{n-1}$ divide the pixels of image f into n classes C_k. Let $T = t_1, t_2, \cdots, t_{n-1}$, then the total Tsallis entropy of n classed $S_q(T)$ is:

$$S_q(T)=\sum_{k=1}^{n} S_q^{C_k}(T)+\sum_{l=1}^{n-1}(1-q)^l S_q^{C_1}(T)S_q^{C_2}(T)\cdots S_q^{Ci+1}(T) \tag{6}$$

where Tsallis entropy of class C_k is:

$$S_q^{C_k}(T)=\frac{1-\sum\limits_{i=t_{k-1}+1}^{t_k}\left(\frac{p_i}{\sum\limits_{i=t_{k-1}+1}^{t_k}p_i}\right)^q}{q-1}, \quad k=1,2,\cdots,n \tag{7}$$

Let $u(t_{k-1}, t_k) = \sum\limits_{i=t_{k-1}+1}^{t_k} p_i$, $w(t_{k-1}+1, t_k) = \sum\limits_{i=t_{k-1}+1}^{t_k} (p_i)^q$, then the sum term in (7) changes into:

$$\sum_{i=t_{k-1}+1}^{t_k}(p_i)^q=\frac{\sum\limits_{i=t_{k-1}+1}^{t_k}(p_i)^q}{u^q(t_{k-1}+1,t_k)}=\frac{w(t_{k-1}+1,t_k)}{u^q(t_{k-1}+1,t_k)} \tag{8}$$

In order to reduce the repetitive computation for Tsallis entropy so as to enhance search domain to search for the optimal thresholds, the recursive equations for computing $u(t_{k-1}+1, t_k)$ and $w(t_{k-1}+1, t_k)$ are given as follows:

$$\begin{cases} w(0,0)=0;\ u(0,0)=0 \\ u(0,t)=\sum\limits_{i=0}^{t}p_i=u(0,t-1)+p_t \\ w(0,t)=\sum\limits_{i=0}^{t}(p_i)^q=w(0,t-1)+p_t^q \\ u(t_{k-1}+1,t_k)=u(0,t_k)-u(0,t_{k-1}) \\ w(t_{k-1}+1,t_k)=w(0,t_k)-w(0,t_{k-1}) \\ k=1,2,\cdots,n \end{cases} \tag{9}$$

When $S_q(T)$ reaches the maximum, the n-1 optimal thresholds can be obtained:

$$(t_1^*, t_2^*, \cdots, t_{n-1}^*) = \underset{0\le t_1<t_2<\cdots<t_{n-1}\le L-1}{Arg\max}\{S_q(T)\} \tag{10}$$

B. BETWEEN-CLASS VARIANCE METHOD

The Otsu based between-class variance method has been employed in determining the optimal thresholding values of an image. The Otsu's method can be described as follows: assume that an image can be represented in L gray

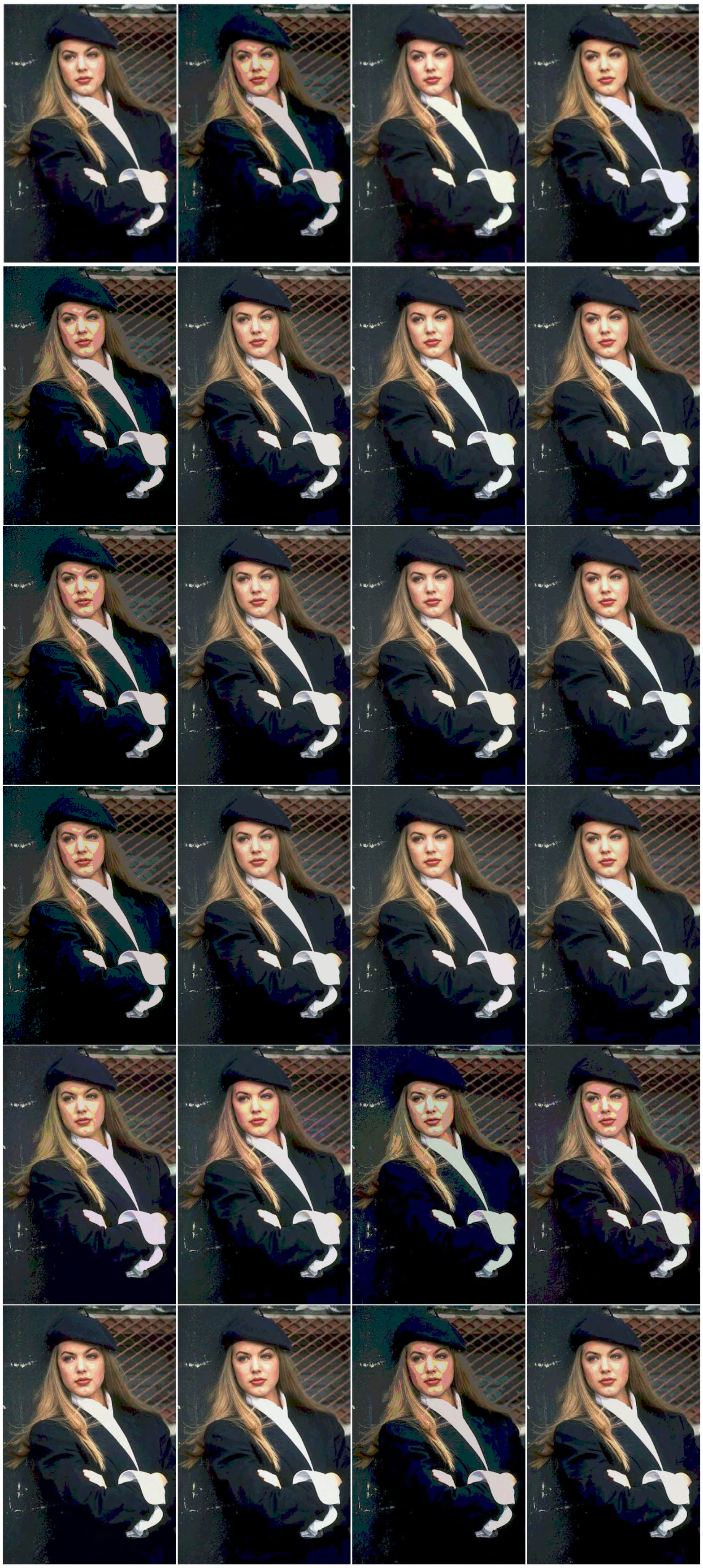

FIGURE 9. The Tsallis results of image test5.

FIGURE 10. The Tsallis results of image test6.

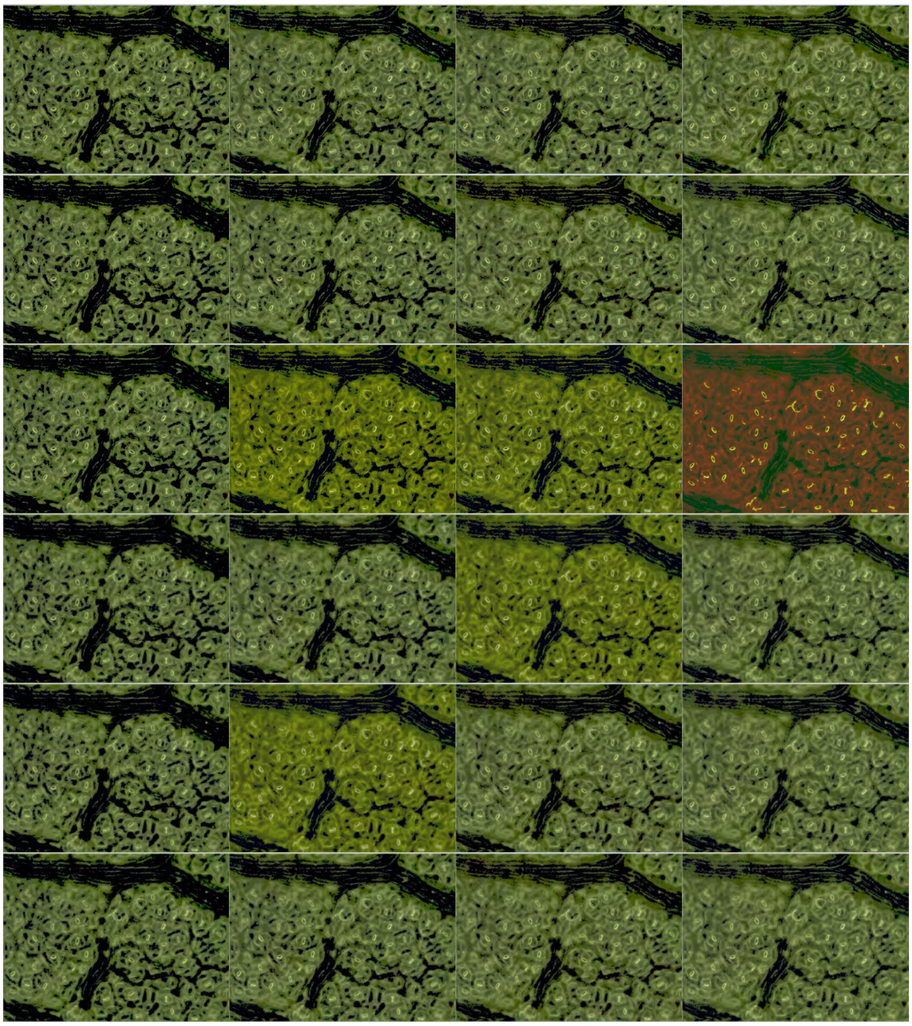

FIGURE 11. The Tsallis results of image Leaf1.

levels $(1, 2, \ldots, L)$ and has N pixels. The number of pixels at level i is denoted by f_1, then $N = f_1 + f_2 + \cdots f_i$. Then, the occurrence probability of gray level i can be defined by the following equation:

$$p_i = \frac{f_i}{N}, p_i \geq 0, \quad \sum_{i=1}^{L} p_i = 1 \tag{11}$$

In bi-level thresholding, the optimum threshold t divides the image into two classes, and the cumulative probabilities of each class can be described as follows:

$$\varpi_0 = \sum_{i=1}^{t} p_i, \quad \varpi_1 = \sum_{i=t+1}^{L} p_i \tag{12}$$

The mean levels of two classes can be defined as follows:

$$\mu_0 = \sum_{i=1}^{t} ip_i/\varpi_0, \quad \mu_1 = \sum_{i=t+1}^{L} ip_i/\varpi_1 \tag{13}$$

Let μ_T be the mean levels of the whole image and it can be defined by

$$\mu_T = \sum_{i=1}^{L} ip_i \tag{14}$$

The between-class variance of whole classes can be represented by

$$f(t) = \sigma_0 + \sigma_1 \tag{15}$$

where $\sigma_0 = \varpi_0(\mu_0 - \mu_T)^2$ and $\sigma_1 = \varpi_1(\mu_1 - \mu_T)^2$. For bi-level thresholding, the Otsu's method find an optimal threshold t^* by maximizing the between-class variance, that is:

$$t^* = \arg\max(f(t)) \tag{16}$$

The Otsu's method can be also extended to multi-level thresholding. Assuming that there are m thresholds, which divide the image into $m + 1$ classes. The extended between-class

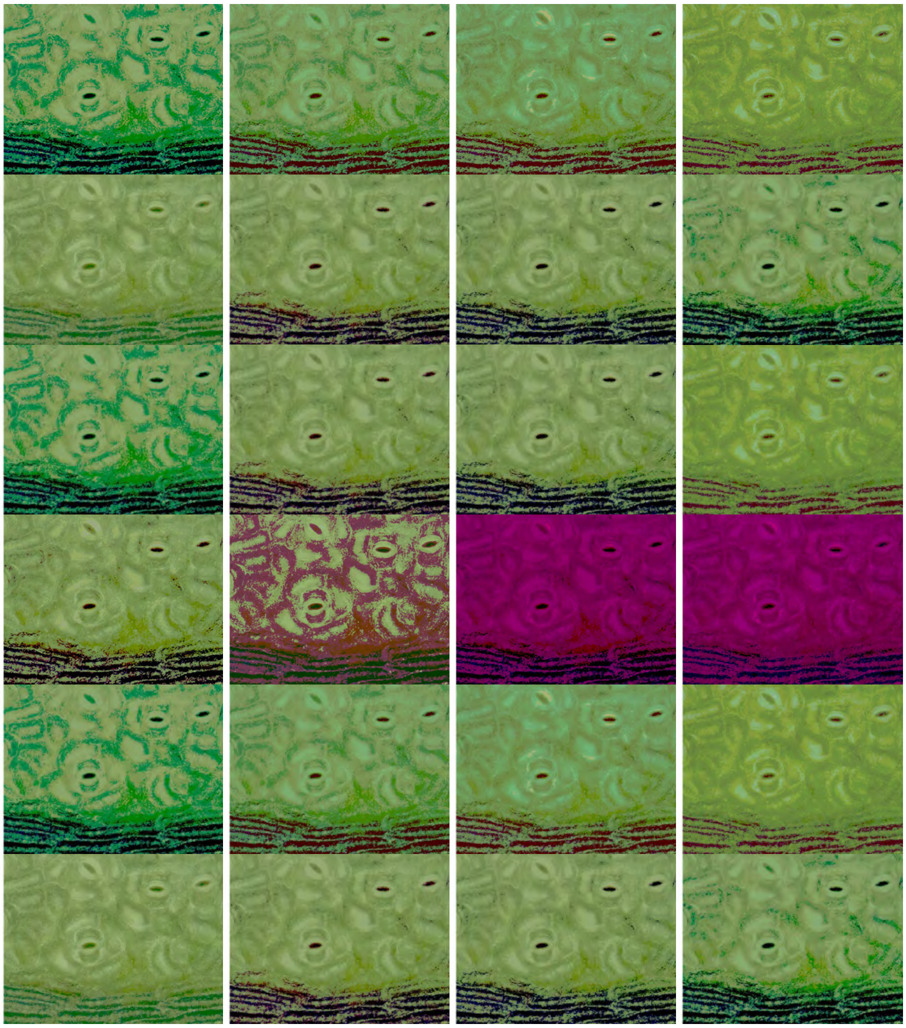

FIGURE 12. The Tsallis results of image Leaf2.

variance is calculated by

$$f(t) = \sum_{i=0}^{m} \sigma_i \tag{17}$$

The sigma terms are determined by (18) and the mean levels are calculated by (19):

$$\sigma_0 = \varpi_0(\mu_0 - \mu_T)^2, \quad \sigma_1 = \varpi_1(\mu_1 - \mu_T)^2, \cdots, \\ \sigma_{M-1} = \varpi_{M-1}(\mu_{M-1} - \mu_T)^2 \tag{18}$$

$$\mu_0 = \sum_{i=1}^{t} ip_i/\varpi_0, \quad \mu_1 = \sum_{i=t_1+1}^{t_2} ip_i/\varpi_1, \cdots, \\ \mu_{M-1} = \sum_{i=t_{M-1}+1}^{L} ip_i/\varpi_{M-1} \tag{19}$$

The optimum thresholds are found by maximizing the between-class variance by (20):

$$t^* = \arg\max(\sum_{i=0}^{M-1} \sigma_i) \tag{20}$$

C. RENYI'S ENTROPY METHOD

Renyi's entropy calculates the absolute value of entropy and the entropy difference in the target region and the background region, then obtains the threshold value t^* in the largest place [52]. The Renyi's entropy of whole classes can be represented by

$$H = H_O + H_B \tag{21}$$

where

$$\begin{cases} H_O(t) = (\frac{1}{1-q}) \sum_{i=0}^{s-1} \sum_{j=0}^{t-1} (\frac{P_{ij}}{P_O})^q \\ H_B(t) = (\frac{1}{1-q}) \sum_{i=s}^{L-1} \sum_{j=t}^{L-1} (\frac{P_{ij}}{P_B})^q \end{cases} \tag{22}$$

where the parameter q is a real number not equal to one associated with the extensitivity of the system, and it is dependent. The threshold value t^* can be found by maximizing:

$$t^* = \arg\max(H_O + H_B) \tag{23}$$

FIGURE 13. The Otsu results of image test1.

Further, Renyi's entropy can be easily extended for the multilevel thresholding problem as given by:

$$H_O(t) = (\frac{1}{1-q})\sum_{i=0}^{s-1}\sum_{j=0}^{t-1}(\frac{P_{ij}}{P_O})^q$$
$$H_1(t) = (\frac{1}{1-q})\sum_{i=s}^{s_1-1}\sum_{j=t}^{t_1-1}(\frac{P_{ij}}{P_B})^q$$
$$H_2(t) = (\frac{1}{1-q})\sum_{i=s_1}^{s_2-1}\sum_{j=t_1}^{t_2-1}(\frac{P_{ij}}{P_B})^q, \cdots$$
$$H_M(t) = (\frac{1}{1-q})\sum_{i=s_m}^{L-1}\sum_{j=t_m}^{L-1}(\frac{P_{ij}}{P_B})^q \tag{24}$$

In order to search M optimal threshold values $[t_1, t_2, \cdots, t_m]$ for a given image, we try to maximize the objective function:

$$t^* = \arg\max(\sum_{i=0}^{m} H_M) \tag{25}$$

III. GRASSHOPPER OPTIMIZATION ALGORITHM

In 2017, Mirjalili Seyedali proposed the grasshopper optimization algorithm. The grasshoppers are a genus of straight fins of insect, they are seen as pests, because they are in crops for food, to cause damage to agriculture. The growth cycle of grasshoppers are shown in Fig.1. The grasshoppers usually exist alone in nature, but they are one of the biggest swarm of all species. The grasshoppers are unique in that they crowd behavior in adults and larvae of between. Millions of larva foraging on the basis of jumping, they feed on almost all plants, and when they reach adulthood, they form a large group in the air, making long migrations, looking for the next food source.

A. MATHEMATICAL MODEL OF GOA

In larvae stage, the main characteristic of grasshopper is moving slowly, small scale food. When they become adult, collective action has became the main activity characteristics of grasshopper. The nature inspired algorithms logically divide the search process into two tendencies: exploration and exploitation [53]. So mathematical model of the gregarious grasshoppers are represented as follows:

$$X_i = S_i + G_i + A_i \tag{26}$$

where X_i defines the position of the *i-th* grasshopper, S_i is the social interaction, G_i is the gravity force on the

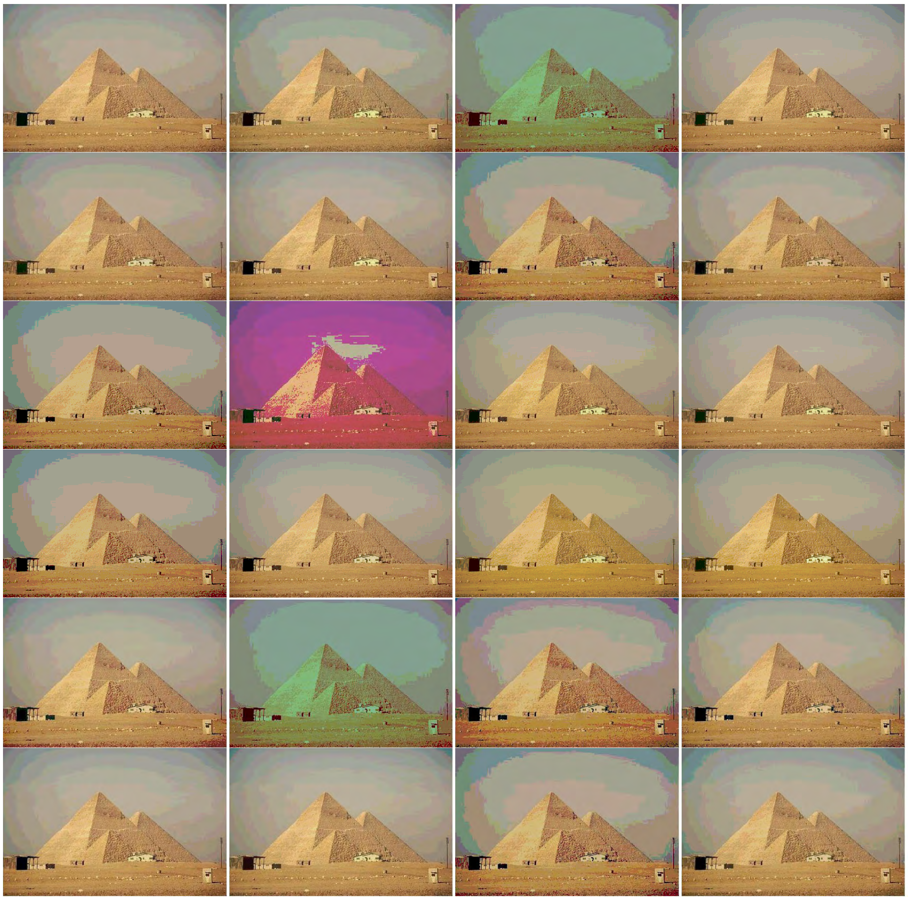

FIGURE 14. The Otsu results of image test2.

i-th grasshopper, and A_i shows the wind advection.

$$\begin{cases} S_i = \sum_{j=1}^{N} s(d_{ij})\vec{d}_{ij} \\ s(r) = fe^{\frac{-r}{l}} - e^{-r} \end{cases} \tag{27}$$

where d_{ij} is the distance between the *i-th* and the *j-th* grasshopper, calculated as $d_{ij} = |x_j - x_i|$, s is a function to define the strength of social forces, $\vec{d}_{ij}$ is an unit vector from the *i-th* grasshopper to the *j-th* grasshopper.

$$G_i = -g\vec{e}_g \tag{28}$$

where g is the gravitational constant and $\vec{e}_g$ shows an unity vector towards the center of earth.

$$A_i = u\vec{e}_w \tag{29}$$

where u is a constant drift and $\vec{e}_w$ is an unity vector in the direction of wind.

Substituting S, G, and A into (26), then this equation can be expanded as follows:

$$X_i = \sum_{j=1}^{N} s(|x_j - x_i|)\frac{x_j - x_i}{d_{ij}} - g\vec{e}_g + u\vec{e}_w \tag{30}$$

However, this mathematical model cannot be used directly to solve optimization problems, mainly because the grasshoppers quickly reach the comfort zone and the swarm does not converge to a specified point. A modified version of this equation is proposed as follows to solve optimization problems:

$$X_i^d = c_1\left(\sum_{j=1}^{N} c_2\frac{ub_d - lb_d}{2}s(|x_j - x_i|)\frac{x_j - x_i}{d_{ij}}\right) + \vec{T}_d \tag{31}$$

Among them, ub_d and lb_d are a type of upper and lower limitation, $\vec{T}_d$ is the optimal value after each iteration, $c_1 = c_2 = c\max - l\frac{c\max - c\min}{L}$, c_1 balances the global search and local search for the target area, c_2 balances the relationship among the attraction between two grasshopper, *cmax* and *cmin* can set the maximum and minimum search ability, l represents the current iteration number, L is the largest number of iterations.

The general framework of GOA-based image thresholding as follows:

B. LEVY FLIGHT TRAJECTORY

Levy's flight was firstly proposed by Levy and then described in detail by Benoit Mandelbrot. In fact, Levy flight is a random step that describes the Levy distribution [54].

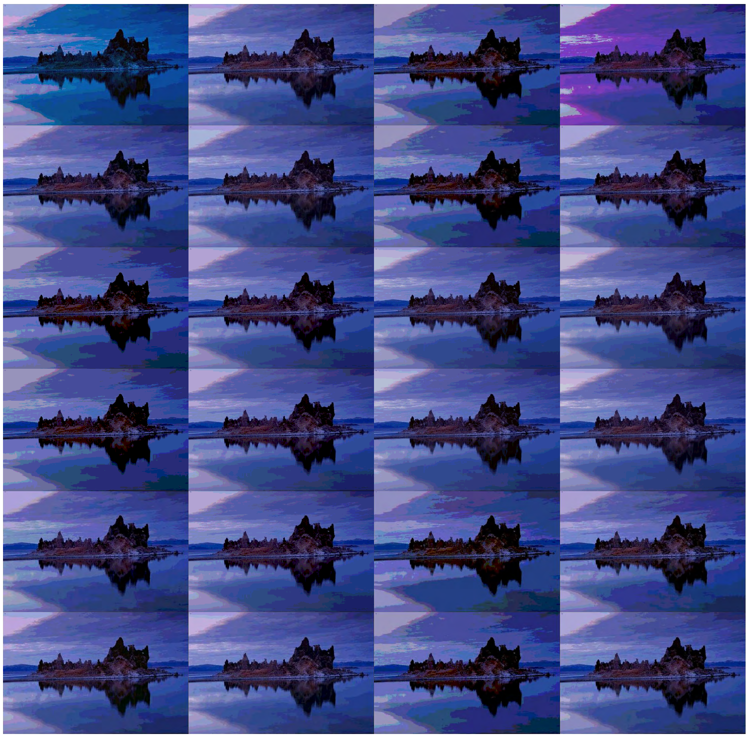

FIGURE 15. The Otsu results of image test3.

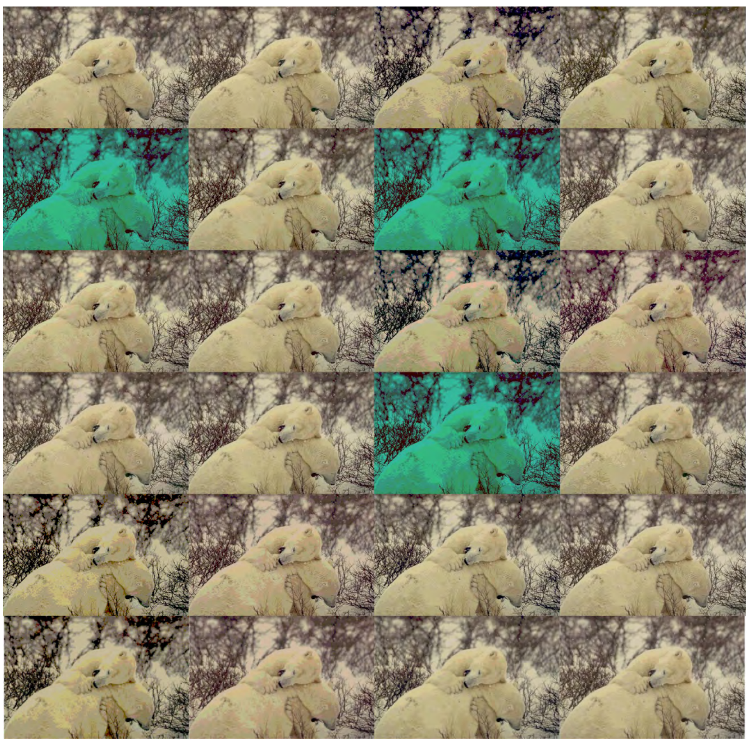

FIGURE 16. The Otsu results of image test4.

Numerous studies have shown that the behavior of many animals and insects are a classic feature of Levy's flight. Levy flight is a special random step method, as shown in Fig.2, which is a simulation of the flight path. Its step length is always small, but occasionally it will also appear large pulsation [55].

FIGURE 17. The Otsu results of image test5.

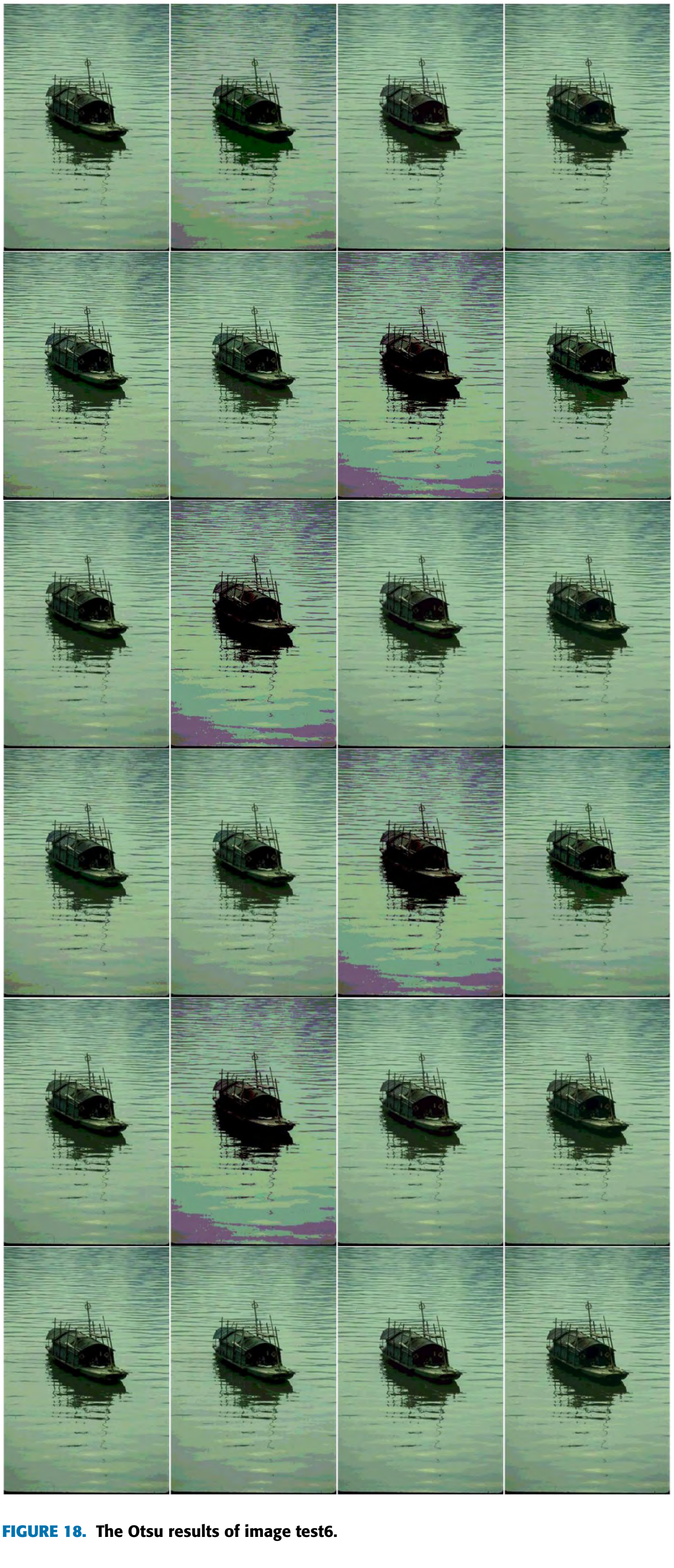

FIGURE 18. The Otsu results of image test6.

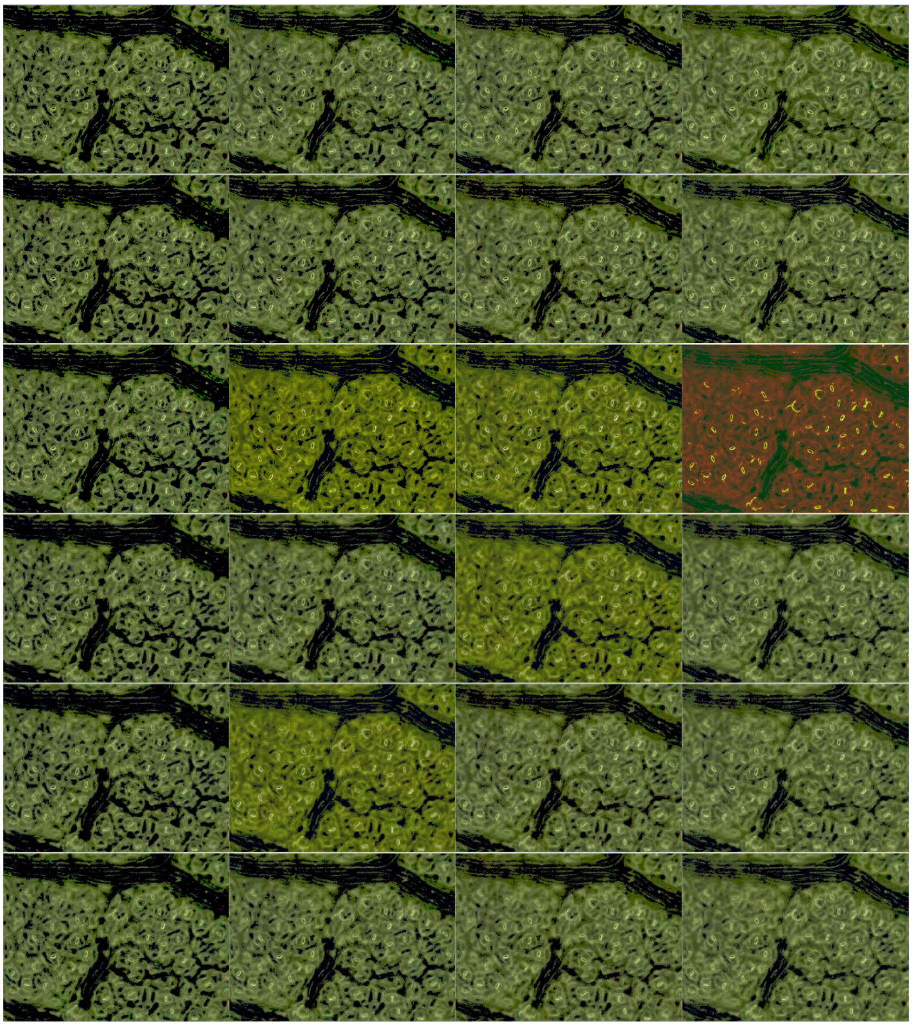

FIGURE 19. The Otsu results of image Leaf1.

The formula for Levy flight is as follows:

$$Levy \sim u = t^{-\lambda}, \quad 1 < \lambda \leq 3 \tag{32}$$

The formula for generating Levy random step proposed by Mantegna is as follows:

$$s = \frac{\mu}{|v|^{1/\beta}} \tag{33}$$

where, parameter $\beta = 1.5$, μ=N(0,σ_μ^2) and v = N(0, σ_μ^2) are gamma functions.

The variance of the parameters is as follows:

$$\sigma_\mu = \left[\frac{\Gamma(1+\beta) \times \sin(\pi \times \beta/2)}{\Gamma[(1+\beta)/2] \times \beta \times 2^{(\beta-1)/2}}\right]^{1/\beta}, \quad \sigma_v = 1 \tag{34}$$

C. MODIFIED GRASSHOPPER OPTIMIZATION ALGORITHM

The GOA algorithm can solve the problem of low dimensional mode optimization simply and efficiently. However, to solve the problem of high and complex image processing, the resulting solution of the traditional GOA is unsatisfactory. In order to improve the global search ability and exploitation ability of GOA, this paper puts forward an improved Levy flight grasshopper optimization algorithm. Levy flight can maximize the implementation of the diversity of the search domain, it can guarantee the algorithm efficient search the location of the function source, realize the local optimum. The findings suggest that Levy flight help GOA get better optimization results. Therefore, the position updating formula of the grasshopper is optimized, and (31) and (32) are improved, which can be expressed by the following mathematical formula:

$$X_i^d = c\left(\sum_{j=1}^{N} c\frac{ub_d - lb_d}{2} s(|x_j - x_i|)\frac{x_j - x_i}{d_{ij}}\right) + Levy * \overrightarrow{T}_d \tag{35}$$

Levy flight can significantly improve the global search ability of GOA to avoid getting into local optimal value. This method not only improves the search of GOA strength,

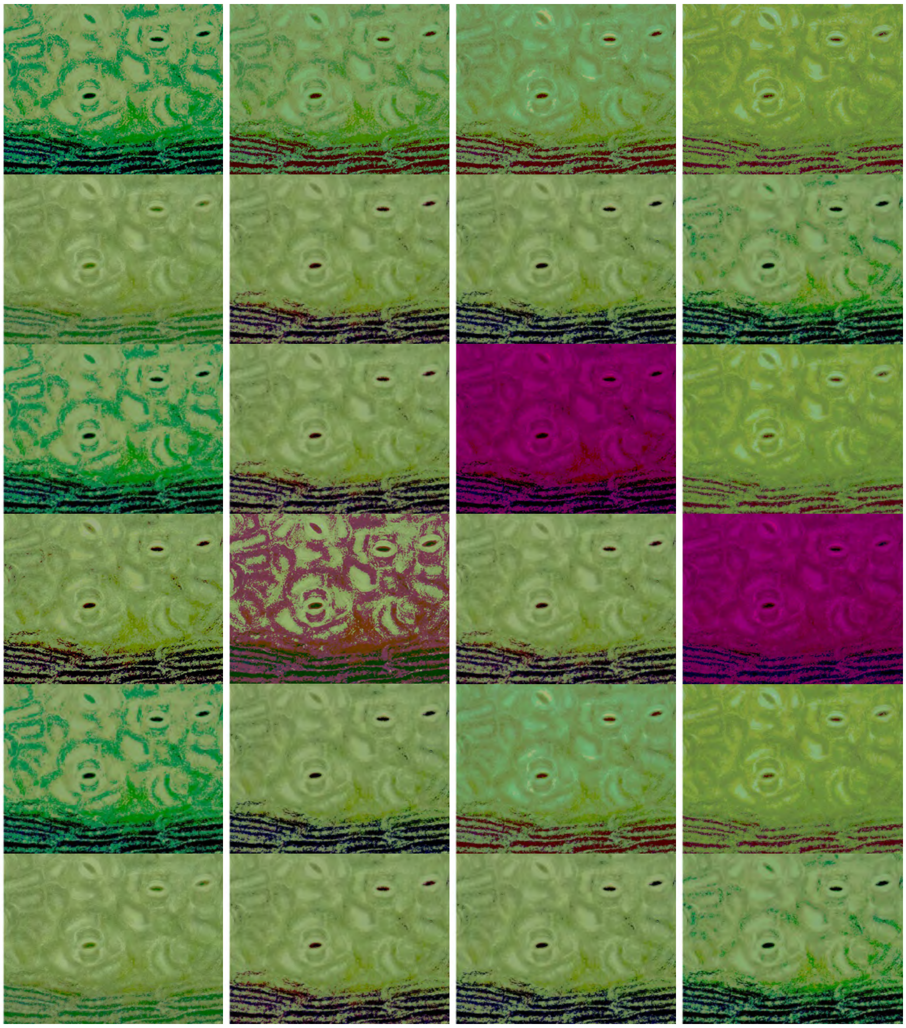

FIGURE 20. The Otsu results of image Leaf2.

but also enhances search domain of the algorithm. Through the optimization algorithm to ensure the algorithm can find the optimal value and avoid falling into local optimum, by enhancing diversity makes the algorithm have better global searching ability. At the same time, the algorithm of single peak and standard functions of multimodal threshold image segmentation has better effect on application. The pseudo code of the MGOA-based image thresholding algorithm is given below.

IV. EXPERIMENTS AND RESULTS

A. EXPERIMENT SETUP

In computer science, mathematics, and management science, optimization is the process of selecting a best solution from some set of available alternatives. In other words, optimization is the method of computing the value of the function and finding the optimal results by maximizing and minimizing an objective function within a given domain. So the objective functions play an important role in the optimization problem. In this paper, Tsallis entropy, between-class variance method and Renyi entropy are using as the objective function that is maximized based on the GOA, WOA, FPA, PSO, BA and MGOA algorithms for multilevel color image thresholding segmentation to find optimal threshold values. The RGB images have three basic color components of red, green and blue, so we need search the optimal threshold values of each component. Moreover, to assess the effectiveness and robustness of the proposed algorithm, recently presented energy curve-based on multilevel thresholding techniques are also investigated. All these compared algorithms are representative algorithms for multilevel thresholding which have been demonstrated in the corresponding references in TABLE 1.

Eight color test images, including natural images and plant stomata images shown in Fig.3 are considered to conduct the experiments. The proposed algorithm is tested over the publicly accessible standard color dataset. Natural color test images are accessed from the Berkeley segmentation data set (BSDS300). The plant stomata images collected by high-power microscopy in the laboratory of Northeast Forestry University. Color images require higher threshold

FIGURE 21. The Renyi results of image test1.

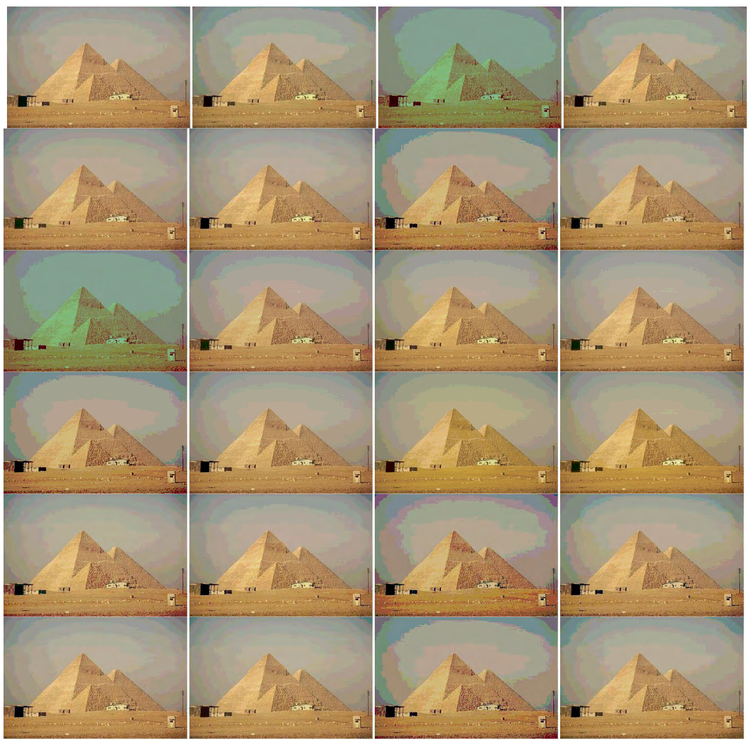

FIGURE 22. The Renyi results of image test2.

levels of segmentation, thus they are more complex to be solved by an optimization technique. Since, optimization algorithms are stochastic and have randomized characteristic. To prevent any discrepancies, all the test images are divided 30 times by the proposed multilevel segmentation algorithm. In TABLE 2-10, the optimal quantitative results

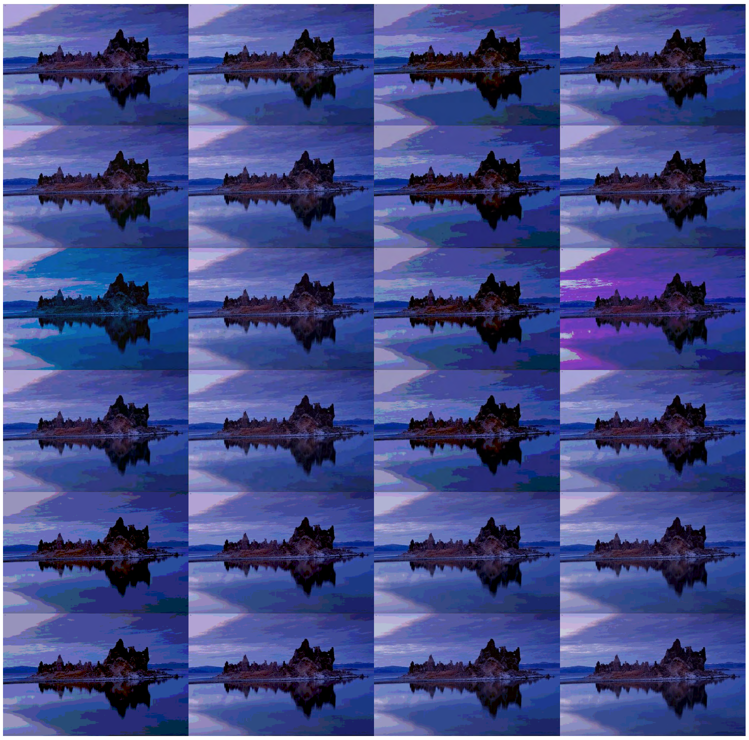

FIGURE 23. The Renyi results of image test3.

Algorithm 1 GOA-Based Image Thresholding

Begin
Initialize the swarm $x_i (i = 1, 2, \ldots, n)$;
Initialize *cmax*, *cmin*, and *maximum number* of *iterations*;
Calculate the fitness of each search agent;
T = the best search agent;
While (l < Max number of iterations)
Update c
for each search agent
Update the position of the current search agent by the (31);
Bring the current search agent back if it goes outside the boundaries;
end for
Update T if there is a better solution;
l = l+1
end while
Return T as the optimal parameter for image thresholding;
End

Algorithm 2 MGOA-Based Image Thresholding

Begin
Initialize the swarm $x_i (i = 1, 2, \ldots, n)$;
Initialize *cmax, cmin*, and *maximum number of iterations;*
T = the best search agent by 10;
While (l < Max number of iterations)
Update c
for each search agent
Update the position of the current search agent by the (31);
Perform Levy flight strategy according to (35);
Bring the current search agent back if it goes outside the boundaries;
end for
Evaluate the fitness of all agents by image thresholding with agent parameters;
Update T if there is a better solution;
l = l + 1
end while
Return T as the optimal parameter for image thresholding;
End

are obtained by simulating the above 30 runs of all test images respectively.

As we know, the value of parameters are very important in determining the performance of swarm intelligence algorithms. So, an extensive set of experiments is conducted to find the right values of the parameters in this paper. Of all the algorithms, the population size is set to be 25 and the max iteration is 500. To demonstrate the superiority of the proposed algorithm, GOA based multilevel thresholding algorithm, including the original GOA and five other popular

FIGURE 24. The Renyi results of image test4.

multilevel thresholding metaheuristic algorithms including GOA, WOA, FPA, PSO and BA algorithm are chosen to compare with the proposed one. PSNR and FSIM values are selected as indicators to evaluate image segmentation results. The parameter of the peak signal to noise ratio (PSNR) is used to compute the peak signal to noise ratio between the original image and the segmented image. The PSNR index is calculated as:

$$PSNR = 20\log(\frac{255}{RMSE})(dB) \tag{36}$$

where

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}\sum_{j=1}^{N}(I(i,j)-\hat{I}(i,j))^2}{M \times N}} \tag{37}$$

where, M, N is the size of the image, I is the original image, and $\hat{I}$ is the segmented image.

The Feature Similarity (FSIM) is used to estimate the structural similarity of the original image and the segmented image. We define FSIM as:

$$FSIM = \frac{\sum_{x\in\Omega} S_L(x)\cdot PC_m(x)}{\sum_{x\in\Omega} PC_m(x)} \tag{38}$$

where Ω represents the entire image, and $S_L(x)$ indicates the similarity between the segmented image obtained through multilevel thresholding task and input image. The FSIM parameter of color RGB image is defined as:

$$FSIM = \frac{1}{O}\sum_{O} FSIM(x^O, y^O) \tag{39}$$

where, x^O and y^O represent oth channel of the original image and segmented image respectively, o is the channel number.

In this experiment, the results obtained by proposing GOA, WOA, FPA, PSO, BA and MGOA algorithms are analyzed at different threshold levels (K = 4,8,10 and 12) for the test images. Table 2 and 3 indicates the threshold value obtained through different approaches using Tsallis entropy threshold for all the color test images. Whereas, PSNR and FSIM values are indicated in Table 4. From the results shown in Table 2 and 3, it can be observed that for all the test images, MGOA significantly produces more favorable and reliable results than GOA, WOA, FPA, PSO and BA especially at high thresholding levels (K) due to their precise search ability. It can be seen from the tabulated values that at T = 3, the algorithm has obtained smaller PSNR, FSIM, while, PSNR obtains higher values and lower RMSE on increasing the

FIGURE 25. The Renyi results of image test5.

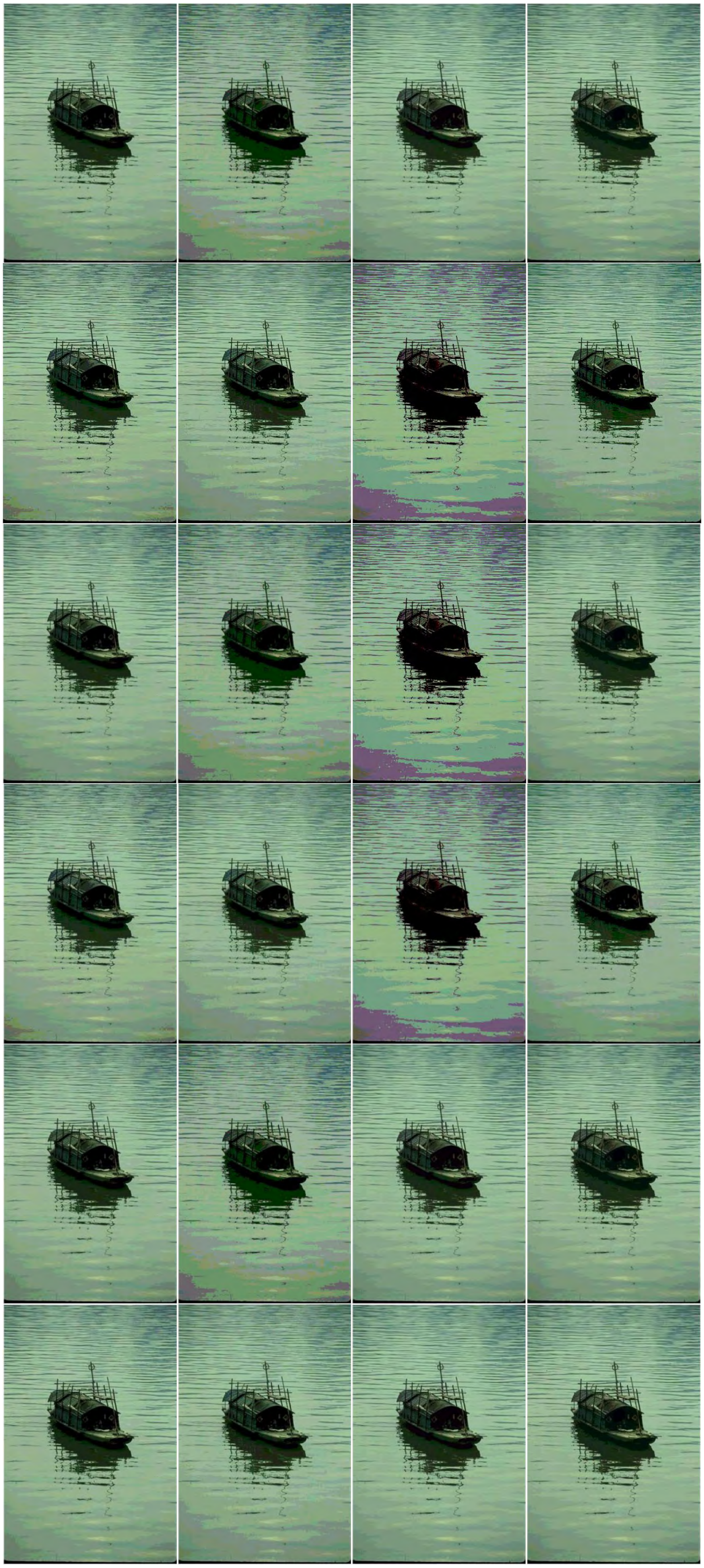

FIGURE 26. The Renyi results of image test6.

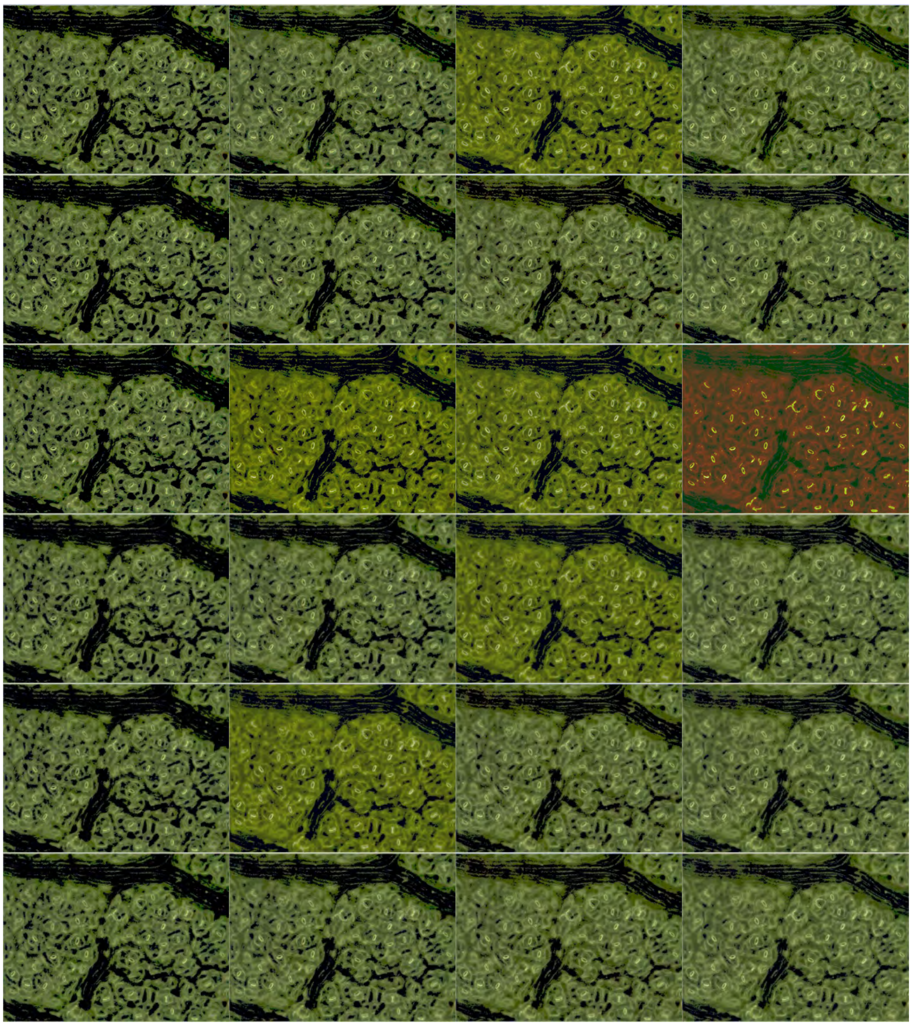

FIGURE 27. The Renyi results of image Leaf1.

threshold levels. Higher PSNR and lower RMSE indicate towards good segmentation quality of the output image.

For a visual, qualitative analysis, the performance of proposed technique at different segmentation levels is represented in Fig.5-10 for colored plant stomata images and in Fig. 11-12 for colored natural images. It can be seen from Fig. 5-12, that the proposed method is able to obtain the satisfying segmentation with well-preserved edges in case of complex plant stomata images as well. To examine, the better quality of optimal threshold values using Tsallis based GOA, WOA, FPA, PSO and BA algorithms.

The objective function values obtained for GOA, WOA, FPA, PSO, BA and MGOA algorithm based on between-class variance are given in TABLE 5-7, respectively. It can be evidently seen from TABLE 5, 6 that the MGOA algorithm has higher objective function values than the GOA, WOA, FPA, PSO and BA algorithm. According to PSNR and FSIM values, MGOA algorithm is superior to other algorithms. The segmentation results for a between-class variance based multilevel segmentation are shown in Fig. 13-20. From a visualization perspective, it can be depicted that in some cases, several pixels are over segmented. Few pixels belonging to the object are divided into background and some of the pixels belonging to the background are divided into objects. It can be depicted that between-class variance based multilevel segmentation approach is not well-suited method for segmentation at lower thresholding levels, since it gives under-segmented and unsuitable outputs. At higher thresholding levels, only MGOA based segmentation has generated fair outcomes.

TABLE 8-10 show the number of thresholds, optimal threshold values, objective functions and their corresponding PSNR and FSIM values obtained by GOA, WOA, FPA, PSO, BA and MGOA algorithm. Segmented images obtained from the MGOA algorithm on the Renyi variance objective function are given in Fig.21-28 for the eight test images. Similarly, comparing PSNR and FSIM values in TABLE 8 and 9, MGOA based techniques have obtained better values than other algorithm based techniques.

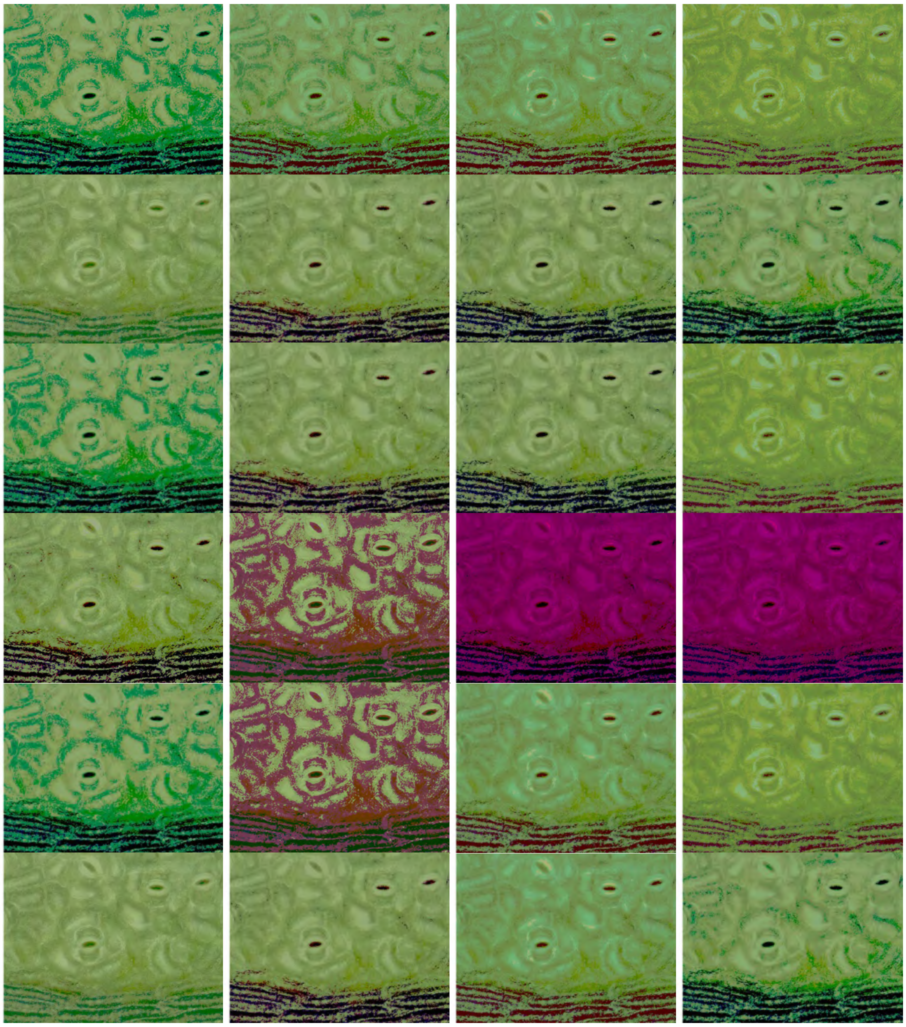

FIGURE 28. The Renyi results of image Leaf2.

From TABLE 2, 3, 5, 6, 8 and 9, it can be found that fitness values of the segmented images by MGOA based Tsallis are more robust than other techniques. It can be clearly seen from TABLE 4, 7 and 10, MGOA has better values and gives higher quality segmentation than other comparable methods such as GOA, WOA, FPA, PSO and BA. It is also seen that the value of FSIM index increases as the amount of thresholds increase. This indicates that segmentation quality improves as the number of thresholds is increased.

B. RESULT

We also perform a statistical analysis of the results. When comparing two methods, we use Wilcoxon rank-sum test [60], a non-parametric statistical test that checks whether one of two independent samples tends to have larger values than the other. The objective function values of the proposed method are compared with GOA, WOA, FPA, PSO and BA based methods. All the algorithms run 30 times for the statistical analysis. Experimental statistical results based on the following TABLE 11-13.

The null hypothesis are constructed as: there is no significant difference between the three algorithms. The alternative hypothesis considers that there is a significant difference between the three algorithms. A value of $p > 0.05$ indicates that the null hypothesis cannot be rejected. On the other hand, a value of $p < 0.05$ means the null hypothesis can be rejected at the 5 % significance level [15]. In the experiment using Tsallis function, MGOA based method produces better result in 28 out of 32 cases when compared with GOA based method and produces better result in 29 out of 32 cases when compared with WOA based method and produces better result in 30 out of 32 cases when compared with FPA based method and produces better result in 32 out of 32 cases when compared with PSO based method and produces better result in 30 out of 32 cases when compared with the BA based method. Whereas, in the experiment using Otsu function, MGOA based method produces better result in 26 out of 32 cases when compared with GOA based method and produces better result in 27 out of 32 cases when compared with WOA based

TABLE 12. P-value of Wilcoxon test comparative Otsu based method.

Images	K	Wilcoxon test				
		MGOA VS GOA	MGOA VS WOA	MGOA VS FPA	MGOA VS PSO	MGOA VS BA
Test1	4	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
	8	P>0.05	P>0.05	P<0.05	P<0.05	P<0.05
	10	P<0.05	P>0.05	P<0.05	P<0.05	P<0.05
	12	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Test2	4	P<0.05	P<0.05	P<0.05	P>0.05	P>0.05
	8	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
	10	P<0.05	P<0.05	P>0.05	P<0.05	P<0.05
	12	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Test3	4	P<0.05	P<0.05	P<0.05	P>0.05	P<0.05
	8	P<0.05	P>0.05	P<0.05	P<0.05	P<0.05
	10	P>0.05	P<0.05	P<0.05	P<0.05	P<0.05
	12	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Test4	4	P<0.05	P>0.05	P<0.05	P<0.05	P<0.05
	8	P<0.05	P<0.05	P>0.05	P>0.05	P<0.05
	10	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
	12	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Test5	4	P<0.05	P<0.05	P>0.05	P<0.05	P>0.05
	8	P>0.05	P<0.05	P<0.05	P>0.05	P<0.05
	10	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
	12	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Test6	4	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
	8	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
	10	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
	12	P>0.05	P<0.05	P<0.05	P<0.05	P<0.05
Leaf1	4	P<0.05	P>0.05	P>0.05	P<0.05	P>0.05
	8	P<0.05	P<0.05	P>0.05	P<0.05	P<0.05
	10	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
	12	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Leaf2	4	P>0.05	P<0.05	P<0.05	P<0.05	P<0.05
	8	P>0.05	P<0.05	P>0.05	P<0.05	P>0.05
	10	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
	12	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05

TABLE 13. P-value of Wilcoxon test comparative Renyi based method.

Images	K	Wilcoxon test				
		MGOA VS GOA	MGOA VS WOA	MGOA VS FPA	MGOA VS PSO	MGOA VS BA
Test1	4	P>0.05	P<0.05	P>0.05	P<0.05	P>0.05
	8	P<0.05	P>0.05	P<0.05	P>0.05	P<0.05
	10	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
	12	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Test2	4	P<0.05	P>0.05	P<0.05	P<0.05	P>0.05
	8	P<0.05	P<0.05	P>0.05	P>0.05	P>0.05
	10	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
	12	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Test3	4	P>0.05	P>0.05	P<0.05	P<0.05	P>0.05
	8	P<0.05	P<0.05	P>0.05	P>0.05	P<0.05
	10	P>0.05	P>0.05	P<0.05	P<0.05	P<0.05
	12	P>0.05	P<0.05	P<0.05	P<0.05	P<0.05
Test4	4	P<0.05	P<0.05	P<0.05	P<0.05	P>0.05
	8	P<0.05	P>0.05	P>0.05	P<0.05	P<0.05
	10	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
	12	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Test5	4	P>0.05	P>0.05	P<0.05	P>0.05	P<0.05
	8	P>0.05	P<0.05	P<0.05	P<0.05	P<0.05
	10	P<0.05	P>0.05	P<0.05	P<0.05	P<0.05
	12	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Test6	4	P>0.05	P<0.05	P>0.05	P>0.05	P>0.05
	8	P>0.05	P<0.05	P>0.05	P<0.05	P>0.05
	10	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
	12	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Leaf1	4	P<0.05	P>0.05	P<0.05	P<0.05	P>0.05
	8	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
	10	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
	12	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Leaf2	4	P<0.05	P<0.05	P>0.05	P<0.05	P>0.05
	8	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
	10	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
	12	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05

method and produces better result in 26 out of 32 cases when compared with FPA based method and produces better result in 28 out of 32 cases when compared with PSO based method and produces better result in 28 out of 32 cases when compared with BA based method. In addition, in the experiment using Renyi function, MGOA based method produces better result in 24 out of 32 cases when compared with GOA based method and produces better result in 24 out of 32 cases when compared with WOA based method and produces better result in 25 out of 32 cases when compared with FPA based method and produces better result in 27 out of 32 cases when compared with PSO based method and produces better result in 23 out of 32 cases when compared with BA based method. It can be seen from results that there is a significant difference between the six algorithms. In most cases MGOA based Tsallis multilevel thresholding algorithm performs better than the other algorithms.

Fig.4 presents the Box-plot representing the PSNR and FSIM for all methods, the MGOA-based multi thresholding image segmentation method can find better threshold values that generate output results with better features. This figure also indicates that many approaches can perform well with a small number of thresholds. This phenomenon suggests that as the number of thresholds increases the complexity of the search space is also significantly incremented.

According to segmentation results, performances of the optimization algorithm are different to use different objective functions. However, on the basis of stability, accuracy, convergence speed, and searching precision, MGOA shows superior performance on multilevel segmentation on energy based Tsallis entropy. WOA and BA have performed fairly, but PSO has an inferior performance on both the objective criteria. Based on the quantitative and qualitative comparison among the presented multilevel segmentation techniques, proposed MGOA technique has achieved good performance as it performs reliable and computationally efficient segmentation through accurate threshold value, selection, and thus has proved to be more suitable for threshold-based image segmentation problems using color images with multimodal distributions such as natural images and plant stomata images at almost all the segmentation levels.

V. CONCLUSIONS

In this paper, a new multilevel thresholding method for color image segmentation based on a modified grasshopper optimization algorithm (MGOA) is proposed. We compare three methods, Tsallis entropy, between-class variables, and Renyi's entropy. Six natural images and two plant stomata images are carried out by using various algorithms, selecting threshold $K = 4,8,10,12$. In order to verify the algorithm is of excellent performance in image segmentation, PSNR and FSIM methods are used. MGOA algorithm in PSNR and FSIM values are higher than other heuristic algorithms and the image segmentation effect is better than other algorithms. As a result, the experiment found the FSIM and PSNR values obtained by Tsallis entropy based segmentation algorithm are superior to other algorithms, which explains the strong ability of Tsallis entropy based algorithm for color image segmentation. As a scope of further research, the MGOA algorithm can be applied to complex applications. In addition, other new objective functions can also be implemented for multilevel color image thresholding segmentation. For other researchers, further work is to be carried out to present an improved grasshopper optimization algorithm for the multilevel image segmentation problem and more complex practical engineering problems.

The result images are the threshold number from left to right, $K = 4,8,10,12$. From top to bottom is GOA, WOA, FPA, PSO, BA and MGOA.

REFERENCES

[1] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.
[2] T. Laux and F. Otto, "Convergence of the thresholding scheme for multi-phase mean-curvature flow," *Calculus Variat. Partial Differ. Equ.*, vol. 55, no. 5, p. 129, 2016.
[3] M. P. van den Heuvel, S. C. de Lange, A. Zalesky, C. Seguin, B. T. T. Yeo, and R. Schmidt, "Proportional thresholding in resting-state fMRI functional connectivity networks and consequences for patient-control connectome studies: Issues and recommendations," *NeuroImage*, vol. 152, pp. 437–449, May 2017.
[4] Y. Li, X. Bai, L. Jiao, and Y. Xue, "Partitioned-cooperative quantum-behaved particle swarm optimization based on multilevel thresholding applied to medical image segmentation," *Appl. Soft Comput.*, vol. 56, pp. 345–356, Jul. 2017.
[5] A. Locatelli, M. Gutzeit, and A. Carpentier, "An optimal algorithm for the thresholding bandit problem," in *Proc. Int. Conf. Int. Conf. Mach. Learn. (JMLR)*, 2016, pp. 1690–1698.
[6] I. Mandić *et al.*, "Charge-collection properties of irradiated depleted CMOS pixel test structures," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, vol. 903, pp. 126–133, Sep. 2018.
[7] M. Hashimoto and T. Nakajima, "Development of a remote sensing algorithm to retrieve atmospheric aerosol properties using multiwavelength and multipixel information," *J. Geophys. Res., Atmos.*, vol. 122, no. 12, pp. 6347–6378, 2017.
[8] P. Simon and P. Schneider, "Weak-lensing shear estimates with general adaptive moments, and studies of bias by pixellation, PSF distortions, and noise," *Astron. Astrophys.*, vol. 604, p. A109, Aug. 2016.
[9] B. Yi, A. D. Rapp, P. Yang, B. A. Baum, M. D. King, "A comparison of Aqua MODIS ice and liquid water cloud physical and optical properties between collection 6 and collection 5.1: Cloud radiative effects," *J. Geophys. Res., Atmos.*, vol. 122, no. 8, pp. 4550–4564, 2017.
[10] J. Dai, Y. Li, K. He, J. Sun. (2016). "R-FCN: Object detection via region-based fully convolutional networks." [Online]. Available: https://arxiv.org/abs/1605.06409
[11] O. Krikeb, V. Alchanatis, O. Crane, and A. Naor, "Evaluation of apple flowering intensity using color image processing for tree specific chemical thinning," *Adv. Animal Biosci.*, vol. 8, no. 2, pp. 466–470, 2017.
[12] C. Zhu, Y. Zheng, M. Savvides, and K. Luu. (2017). "CMS-RCNN: Contextual multi-scale region-based CNN for unconstrained face detection." [Online]. Available: https://arxiv.org/abs/1606.05413
[13] L. Shen, X. Huang, and C. Fan, "Double-group particle swarm optimization and its application in remote sensing image segmentation," *Sensors*, vol. 18, no. 5, pp. 1393–1394, 2018.
[14] E. Tuba, A. Alihodzic, and M. Tuba, "Multilevel image thresholding using elephant herding optimization algorithm," in *Proc. IEEE Int. Conf. Eng. Mod. Electr. Syst.*, Jun. 2017, pp. 240–243.
[15] A. K. M. Khairuzzaman and S. Chaudhury, "Multilevel thresholding using grey wolf optimizer for image segmentation," *Expert Syst. Appl.*, vol. 86, pp. 64–76, Nov. 2017.
[16] S. Agrawal, R. Panda, and A. Abraham, "A novel diagonal class entropy-based multilevel image thresholding using coral reef optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 1, pp. 1–9, 2018.
[17] A. R. Plastino and A. Plastino, "Stellar polytropes and Tsallis' entropy," *Phys. Lett. A*, vol. 174, pp. 384–386, Mar. 1993.
[18] S. Wang *et al.*, "Pathological brain detection via wavelet packet tsallis entropy and real-coded biogeography-based optimization," *Fundam. Inform.*, vol. 151, nos. 1–4, pp. 275–291, 2017.

[19] S. J. Mousavirad and H. Ebrahimpour-Komleh, "Multilevel image thresholding using entropy of histogram and recently developed population-based metaheuristic algorithms," *Evol. Intell.*, vol. 10, nos. 1–2, pp. 45–75, 2017.
[20] N. S. M. Raja, P. R. V. Lakshmi, and K. P. Gunasekaran, "Firefly algorithm-assisted segmentation of brain regions using Tsallis entropy and Markov random field," in *Innovations in Electronics and Communication Engineering*. Singapore: Springer, 2018.
[21] A. Alva, R. S. Akash, and K. Manikantan, "Optimal multilevel thresholding based on Tsallis entropy and half-life constant PSO for improved image segmentation," in *Proc. IEEE Elect. Comput. Electron.*, Dec. 2016, pp. 1–6.
[22] S. Sarkar, S. Das, and S. S. Chaudhuri, "Hyper-spectral image segmentation using Rényi entropy based multi-level thresholding aided with differential evolution," *Expert Syst. Appl.*, vol. 50, pp. 120–129, May 2015.
[23] L. He and S. Huang, "Modified firefly algorithm based multilevel thresholding for color image segmentation," *Neurocomputing*, vol. 240, pp. 152–174, May 2017.
[24] N. Muangkote, K. Sunat, and S. Chiewchanwattana, "Multilevel thresholding for satellite image segmentation with moth-flame based optimization," in *Proc. IEEE Int. Joint Conf. Comput. Sci. Softw. Eng.*, Jul. 2016, pp. 1–6.
[25] S. Pare, A. K. Bhandari, V. Bajaj, and A. Kumar, "Backtracking search algorithm for color image multilevel thresholding," *Signal Image Video Process.*, vol. 12, no. 2, pp. 385–392, 2018.
[26] V. K. Bohat and K. V. Arya, "A new heuristic for multilevel thresholding of images," *Expert Syst. Appl.*, vol. 117, pp. 176–203, Mar. 2019.
[27] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," *J. Electron. Imag.*, vol. 13, no. 1, pp. 146–166, 2004.
[28] R. Gao *et al.*, "Human infection with a novel avian-origin influenza a (H7N9) virus," *New England J. Med.*, vol. 368, no. 20, pp. 1888–1897, 2013.
[29] S. Pare, A. K. Bhandari, A. Kumar, and G. K. Singh, "An optimal color image multilevel thresholding technique using grey-level co-occurrence matrix," *Expert Syst. Appl.*, vol. 87, pp. 335–362, Nov. 2017.
[30] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed optimization by ant colonies," in *Proc. 1st Eur. Conf. Artif. Life*, vol. 142, 1991, pp. 134–142.
[31] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micro Mach. Hum. Sci.*, 1995, pp. 39–43.
[32] M. Maitra and A. Chatterjee, "Hybrid multiresolution Slantlet transform and fuzzy *c*-means clustering approach for normal-pathological brain MR image segregation," *Med. Eng. Phys.*, vol. 30, no. 5, pp. 615–623, 2008.
[33] H. Gao, W. Xu, Y. Tang, and J. Sun, "Multilevel thresholding for image segmentation through an improved quantum-behaved particle swarm algorithm," *IEEE Trans. Instrum. Meas.*, vol. 59, no. 4, pp. 934–946, Apr. 2010.
[34] P. Ghamisi, M. S. Couceiro, J. A. Benediktsson, and F. M. L. Martins, "Multilevel image segmentation based on fractional-order darwinian particle swarm optimization," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2382–2394, May 2014.
[35] D. Karaboga and B. Basturk, "Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems," in *Proc. 12th Int. Fuzzy Syst. Assoc. World Congr. Found. Fuzzy Logic Soft Comput.*, 2007, pp. 789–798.
[36] J. Liu, J. Li, and Y. Zhao, "The occurrence of perfluorinated alkyl compounds in human milk from different regions of China," *Environment Int.*, vol. 36, no. 5, pp. 433–438, 2010.
[37] X.-S. Yang, "Flower pollination algorithm for global optimization," in *Proc. Int. Conf. Unconventional Comput. Natural Comput.*, 2013, pp. 240–249.
[38] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," *Nature Inspired Cooperat. Strategies Optim.*, vol. 284, pp. 65–74, 2010.
[39] X. Cai, X.-Z. Gao, and Y. Xue, "Improved bat algorithm with optimal forage strategy and random disturbance strategy," *Int. J. Bio-Inspired Comput.*, vol. 8, no. 4, pp. 205–214, 2016.
[40] C. Li, Z.-Y. Liu, X. Yang, H. Qiao, and J.-H. Su, "Stitching contaminated images," *Neuro Comput.*, vol. 214, pp. 829–836, Nov. 2016.
[41] C. Karri and U. Jena, "Fast vector quantization using a Bat algorithm for image compression," *Eng. Sci. Technol., Int. J.*, vol. 19, pp. 769–781, Jun. 2016.
[42] G.-G. Wang, H. E. Chu, and S. Mirjalili, "Three-dimensional path planning for UCAV using an improved bat algorithm," *Aerosp. Sci. Technol.*, vol. 49, pp. 231–238, Feb. 2016.
[43] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.
[44] M. M. Mafarja and S. Mirjalili, "Hybrid whale optimization algorithm with simulated annealing for feature selection," *Neuro Comput.*, vol. 260, pp. 302–312, Oct. 2017.
[45] I. Aljarah, H. Faris, and S. Mirjalili, "Optimizing connection weights in neural networks using the whale optimization algorithm," *Soft Comput.*, vol. 22, no. 1, pp. 1–15, 2016.
[46] K. B. O. Medani, S. Sayah, and A. Bekrar, "Whale optimization algorithm based optimal reactive power dispatch: A case study of the Algerian power system," *Electr. Power Syst. Res.*, vol. 163, pp. 696–705, Oct. 2017.
[47] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.
[48] H. Haklı and H. Uğuz, "A novel particle swarm optimization algorithm with Levy flight," *Appl. Soft Comput.*, vol. 23, pp. 333–345, Oct. 2014.
[49] S. Amirsadri, S. J. Mousavirad, and H. Ebrahimpour-Komleh, "A Levy flight-based grey wolf optimizer combined with back-propagation algorithm for neural network training," *Neural Comput. Appl.*, vol. 30, no. 12, pp. 3707–3720, 2017.
[50] S. J. Mousavirad and H. Ebrahimpour-Komleh, "Human mental search: A new population-based metaheuristic optimization algorithm," *Appl. Intell.*, vol. 47, no. 3, pp. 850–887, 2017.
[51] K. Tang, X. Xiao, J. Yang, L. Luo, and J. Wu, "An improved multilevel thresholding approach based modified bacterial foraging optimization," *Appl. Intell.*, vol. 46, no. 1, pp. 214–226, 2017.
[52] S. Sarkar, N. Sen, A. Kundu, S. Das, and S. S. Chaudhuri, "A differential evolutionary multilevel segmentation of near infra-red images using Renyi's entropy," in *Proc. Int. Conf. Frontiers Intell. Comput., Theory Appl. (FICTA)*. Berlin, Germany: Springer, 2013.
[53] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm: Theory and application," *Adv. Eng. Softw.*, vol. 105, pp. 30–47, Mar. 2017.
[54] J. H. Santillan, S. Tapucar, and C. Manliguez, "Cuckoo search via Lévy flights for the capacitated vehicle routing problem," *J. Ind. Eng. Int.*, vol. 14, pp. 293–304, Aug. 2017.
[55] S. Gattenlöhner, I. V. Gornyi, B. Trauzettel, A. D. Mirlin, M. Titov, and P. M. Ostrovsky, "Lévy flights due to anisotropic disorder in graphene," *Phys. Rev. Lett.*, vol. 117, no. 4, p. 046603, 2016.
[56] M. A. El Aziz, A. A. Ewees, and A. E. Hassanien, "Whale optimization algorithm and moth-flame optimization for multilevel thresholding image segmentation," *Expert Syst. Appl.*, vol. 83, pp. 242–256, Oct. 2017.
[57] J. Xue, X. He, X. Hao, F. He, and X. Yang, "Multi-threshold image segmentation method based on flower pollination algorithm," in *Bio-inspired Computing: Theories and Applications*. Singapore: Springer, 2017.
[58] T. X. Pham, P. Siarry, and H. Oulhadj, "Integrating fuzzy entropy clustering with an improved PSO for MRI brain image segmentation," *Appl. Soft Comput.*, vol. 65, pp. 230–242, Apr. 2018.
[59] Z.-W. Ye, M.-W. Wang, S.-B. Chen, and W. Liu, "Fuzzy entropy based optimal thresholding using bat algorithm," *Appl. Soft Comput.*, vol. 31, pp. 381–395, Jun. 2015.
[60] P. Mesejo, A. Valsecchi, L. Marrakchi-Kacem, S. Cagnoni, and S. Damas, "Biomedical image segmentation using geometric deformable models and metaheuristics," *Comput. Med. Imag. Graph.*, vol. 43, pp. 167–178, Jul. 2015.

HONGNAN LIANG was born in Daqing, China, in 1996. She is currently pursuing the B.S. degree in electrical engineering and automation with Northeast Forestry University, China. Her research interests include image segmentation and intelligent optimization algorithms.

HEMING JIA received the Ph.D. degree in system engineering from Harbin Engineering University, China, in 2012. He is currently an Associate Professor with Northeast Forestry University. His research interests include nonlinear control theory and application, image segmentation, and swarm optimization algorithms.

JUN MA was born in Yichang, China, in 1999. He is currently pursuing the B.S. degree in automation with Northeast Forestry University, China. His research interests include image segmentation and intelligent optimization algorithms.

ZHIKAI XING was born in Daqing, China, in 1993. He is currently pursuing the M.S. degree in control engineering with Northeast Forestry University, China. His research interests include image segmentation and intelligent optimization algorithms.

XIAOXU PENG was born in Harbin, China, in 1995. He is currently pursuing the M.S. degree in control theory and control engineering with Northeast Forestry University, China. His research interests include image segmentation and intelligent optimization algorithms.