A Novel Codebook Generation by Smart Fruit Fly Algorithm based on Exponential Flight

Ilker Kilic
Electrical and Electronics Engineering Department,
Pamukkale University, Turkey
ilkerk@pau.edu.tr

Abstract: *A codebook is a combination of vectors that represents a digital image best and very useful tool for compression. Besides the well-known techniques such as Linde-Buzo-Gray, C-Means, and Fuzzy C-Means the nature-inspired metaheuristic algorithms have also become alternate techniques for solving the codebook generation problem. Fruit Fly Optimization Algorithm (FFA) is a simple and efficient algorithm, but the capturing of an agent by a local minimum point is the main problem. Therefore, the fruit flies generally do not reach the global solution at the end of the iterations. In this study, the FFA is empowered with a smart exponential flight approach to finding out a global optimum codebook. In this approach, if a fruit fly agent is captured by a local minimum point accidentally, the smart exponential flight steps provide an opportunity to escape from it easily. In the experimental studies, successful compression results have been taken in terms of lower error rates. The numerical results prove that the proposed Smart Exponential flight-based Fruit Fly Algorithm (SE-FFA) is better than the variations of convolutional FFA by providing a global optimum codebook.*

Keywords: *Metaheuristic optimization technique, fruit fly algorithm, image compression, codebook generation.*

Received July 7, 2021; accepted April 13, 2022
https://doi.org/10.34028/iajit/20/4/4

1. Introduction

Vector Quantization (VQ) is a lossy image encoding algorithm combined by the Linde-Buzo-Gray technique (LBG) that determines the codebook [8, 18, 19, 21, 27]. VQ contains three sub procedures called codebook generation, compression and decoding. The codebook generation algorithm is so important that it directly determines the quality of the reconstructed image. With the development of microprocessor technology, different metaheuristic algorithms have also been used for codebook generation.

The Genetic Algorithm (GA) is a branch of evolutionary algorithms that simulates the natural evolution process to overcome the optimization problem. GA is used for codebook generation [9, 22] and combined with different techniques to generate better codebooks [26, 30]. Ant Colony Optimization (ACO) [12, 28], Particle Swarm Optimization (PSO) [5], Honey Bee Mating Optimization (HBMO) [7], Firefly Optimization (FRO) [24], Tabu Search Algorithm (TSA) [23], Bat optimization algorithm (BAT) [13] and Cuckoo Search Optimization (CSO) [2, 12] are also combined with VQ technique for codebook generation.

Fruit Fly Optimization Algorithm (FFA) is an optimization technique that imitates the drosophila fly [3]. The FFA is an easy bio-inspired algorithm to understand and has a simple computational flow chart. Hence, the FFA technique has been implemented in different kinds of optimization problems. Some of them can be listed as financial distress models [3], the hybrid annual power load forecasting [11], satisfaction determination in logistics task using both fruit fly optimization and neural network, annual electric load forecasting [15], adjusting PID controller [14, 25], artificial neural network [1], knapsack problem [20], determination the parameters of synchronous generator [31], problems of the joint replenishment [29], FFA based Fuzzy-PID controller [25], solving the steelmaking casting problem [16]. Recently, FFA has been specifically implemented in image watermarking using a deep learning method through wavelet transform [10]. Nowadays FFA is applied to different engineering problems [4, 6, 17, 28, 32].

Although FFA is an efficient and fast metaheuristic approach, the constant radius generally prevents the agents from escaping the local minimum points. In this new proposed study, two new approaches are added into the convolutional FFA. The first one is a new robust initial codebook which is prepared by a statistical analysis to stretch the color space of the image blocks. The second one is determining the smart exponential radius functions of the fruit flies. The smart exponential radius depends on the standard deviation and coefficient of variation Equation (9) of the input image which is calculated by a preprocess. The coefficient of variation value is used to define if the image is low or high contrast. If the image is high contrast, it is difficult to process and needs extra iteration steps to find out the

global codebook. Therefore, the slope of the exponential function is determined as low. On the contrary, if the image is low contrast, it is easier to represent the image with a fewer number of codewords and needed fewer iteration steps. Therefore, the slope of the exponential function is determined as high. With the help of these two newly introduced steps, the smart fruit flies always start to fly from better positions and reach the global minimum point easily.

The paper is composed of five sections. Following the introduction, the popular quantization algorithms are explained briefly in section 2. The proposed SE-FFA for codebook generation is introduced in section 3. The parameters and the experimental results are explained in section 4. A conclusion is declared in section 5.

2. Related Works

In this section, VQ and the standard FFA algorithm is presented briefly.

2.1. Vector Quantization and LBG Algorithm

Vector Quantization (VQ) is an image compression algorithm. The codebook producing is very important part of VQ. Let the size of input image Y={x_{ij}} be N×N pixels size and represented by sub blocks with the size of m×m pixels. Therefore, the whole image is defined by Nb= $\left(\frac{N}{m} \times \frac{N}{m}\right)$ sub blocks which are determined by a group of original image vectors X=xi and i=1, 2, …, Nb. Let L be a value assumed as m×m. The original sub blocks xi defined as xi∈ℜL that ℜL is named as Euclidean space with L dimensional. A codebook is collection of Nc codewords, C={C_1,C_2,…,C_{Nc}}, j=1,2,..,N_c, Cj ∈ ℜL. Original image vectors are represented by a row vector x_i=(x_{i1}, x_{i2}, x_{i3}, …, x_{iL}) and the i^{th} codeword is defined as c_i=(c_{i1}, c_{i2}, x_{i3}, …, c_{iL}). In VQ, the most suitable codeword from the codebook is assigned to each image block. Therefore, instead of using the original image block the number of the codeword from the codebook is used for the purpose of lossy compression as seen in Figure 1. Determination of the C is achieved by minimizing the following Mean Square Error (MSE) criteria,

$$\mathrm{MSE(C)} = \frac{1}{N_b}\sum_{j=1}^{N_c}\sum_{i=1}^{N_b}\mu_{ij}\left\|x_1 - c_j\right\|^2 \quad (1)$$

$$\sum_{j=1}^{N_c}\mu_{ij} = 1\,,\ i \in \{1,2,\dots,N_b\} \quad (2)$$

$$\mu_{ij} = \begin{cases} 1, \text{if } x_i \text{ is in the } j^{th}\text{cluster} \\ 0, \qquad \text{otherwise} \end{cases} \quad (3)$$

The $\|x_i - c_j\|$ is the Euclidean margin between the codeword c_j and vector x_i. Following two rules exist for a local optimum codebook.

a) The group of vector (R_j, j=1,2, … N_c) must satisfy

$$R_j \supset \{x \in X: d(x,c_j) < d(x,c_k)\,,\ \forall k \neq j\} \quad (4)$$

b) The center of the R_j called c_j is calculated as

$$c_j = \frac{1}{N_j}\sum_{i=1}^{N_j} x_i\,,\ \ x_i \in R_j \quad (5)$$

Here, N_j is the total number of elements belongs to R_j. Let the image vectors, x_i, i=1,2, …, N_b, Euclidean distance d, and starting codewords $c_j(0)$, j=1, 2,…,N_c. The LBG technique executes the following three steps in order to obtain the local optimal codebook:

a) Partition the original image blocks into several clusters using Euclidean distance. The result of clustering is saved in a N_b×N_c sized matrix named U. Elements of U are determined as

$$\mu_{ij} = \begin{cases} 1, \text{if } \ d\left(x_i, c_j(k)\right) = \min d\left(x_i, c_j(k)\right) \\ 0, \qquad\qquad\qquad \text{otherwise} \end{cases} \quad (6)$$

b) Define the center of each cluster. Replace the old centers by the following new ones.

$$c_j(k+1) = \frac{\sum_{i=1}^{N_b}\mu_{ij}x_i}{\sum_{i=1}^{N_b}\mu_{ij}},\ j = 1,2,\dots,N_c \quad (7)$$

Repeat the steps *a* and *b* until there is no change of centroid c_j values.

2.2. Convolutional FFA

The FFA is an optimization algorithm, which imitates the food behavior of fruit fly called Drosophila [30]. A fruit fly follows two behaviors for food. First one smells food source by using its osphresis sensor, and goes to that location. Then the drosophila uses its sensitive vision feature in order to determine if there is a better food source by looking around to the other fruit flies. Then all drosophilas fly toward the best food location using their vision feature by leaving their food storage. Fruit fly foraging procedure can be grouped into three parts;

1. Initial population generation, parameter setting.
2. Osphresis search step.
3. Vision search step.

The food search steps of FFA are illustrated in Figure 1 and the steps of the algorithm are given as;

- *Step* 1: Determine the iteration number, search diameter and number of fruit fly.
- *Step* 2: Determine the locations of the fruit flies randomly.
- *Step* 3: Osphresis search phase: randomly food search of the fruit flies in the predefined search location.
- *Step* 4: Define the smell density of fruit flies.
- *Step* 5: Search phase by vision: define the fruit fly which has the maximum smell density. Then, all fruit flies go to the best smell location. In the next iteration, this location becomes the new swarm location.
- *Step* 6: Stop FFA algorithm if the last iteration is executed, otherwise go to the steps of 3, 4, 5, and 6 sequentially.

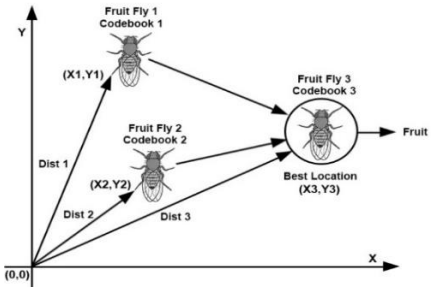


Figure 1. The foraging process of a fruit fly swarm.

3. Smart Exponential Flight Based Fruit Fly Algorithm

The random initialization of the classical FFA makes the algorithm unstable. If the best smell of the initial fruit fly population is not good or the diameter of the osphresis search phase is not chosen suitably, the fruit flies are generally captured by a local minimum location. Therefore, in this paper a new robust initialization technique is combined by a smart exponential diameter search osphresis phase in order to overcome this problem to reach the global minimum.

3.1. A New Robust Initialization Procedure

In the VQ based FFA, the initial codebook generation strategy is so important that it directly affects the rest of the FFA performance. More flies start from better locations to reach the best food easily. So, a robust initial codebook production that stretches the color space is required at the beginning of the algorithm. The new introduced initialization technique that stretches the color space provides reaching to the global optimum solution target robustly. For this purpose, the original image of 256×256 pixel size is divided into4096 subblocks of 4×4 pixel size. Then all 4096 blocks are reordered according to its mean value and grouped for random codeword selection as seen in Figure 2.

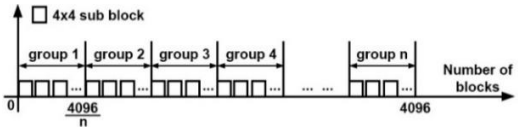


Figure 2. Reordering the image sub blocks according to their average gray levels.

Each single codebook is composed of n codewords. Consequently, 4096 averagely ordered blocks are divided into n groups, each containing 4096/n codewords. As a result, each codeword of the initial codebook is randomly selected from the corresponding group. Since the average gray levels of codeword groups are Increased from 0 to 255 gray levels, then a composed codebook stretches the gray level space.

3.2. Image Compression with Fruit Fly

Codebook design for VQ is the main process of the proposed algorithm seen in Figure 3. Since the problem requires excessive calculations and comparisons, the number of fruit flies is selected as 100. Each Fruit fly represents a unique codebook. First of all, before FFA, the new proposed initial codebook production method is applied, seen in Figure 2. Instead of sending all gray pixel values in a block, the number of best matching in the codebook is sent. Thus, the less codeword in a codebook means higher compression ratio is achieved.

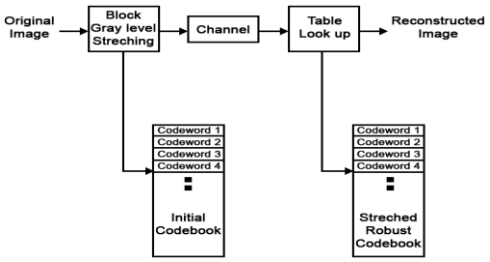


Figure 3. Image compression procedure by proposed SE-FFA.

3.3. Smart Exponential Function Search Strategy

In order to increase the performance of standard FFA one of the choices is using variable search radius. When a fruit fly begins foraging, the search radius must be large enough in order to increase the possibility of food. But when a fruit fly finds a trace of food, the search area must be selected as a smaller one. In this proposed SE-FFA, an exponential function based search diameter is used as in Equation (8).

$$R(N) = R_{max} \cdot e^{(aN)} \quad (8)$$

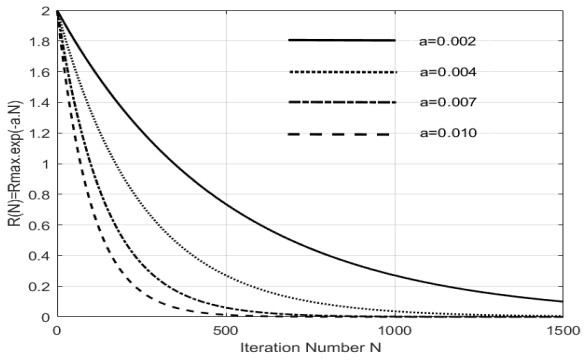


Figure 4. Exponential function slopes according to a, N, R(N).

Where the variable *a* is the slope of exponential function, N is the iteration number. As seen in Figure 4, drosophila has a long radius at the beginning of the search while it decreases to zero by the end of iteration.

In this proposed technique, the coefficient of variation (cov) metric (Equation (9)) is used for selection of the constant value a. The cov is a normalized metric of variation. In this work, it represents the normalized gray level variation of codewords in the codebook.

$$COV = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(X_i\text{-mean})^2}}{\text{mean}} \quad (9)$$

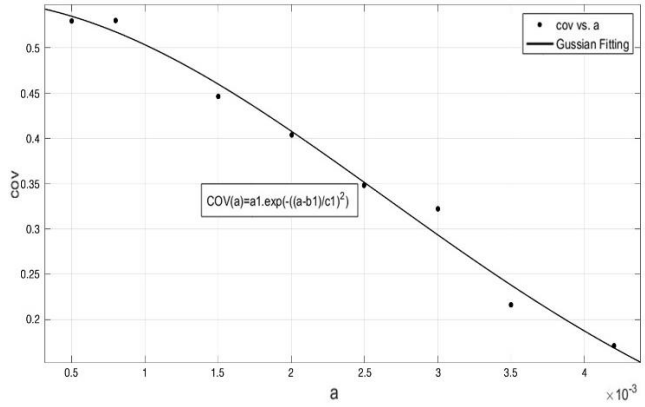


Figure 5. Gaussian function curve fitting.

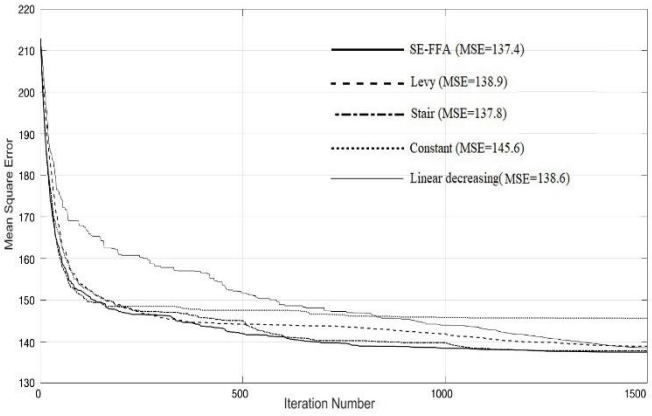


Figure 6. Original 256×256 pixel size of low contrast *moon* image with COV=0.217. SE-FFA convergence results with the compression ratio of 0.122 bpp and performance comparison of different radius types starting from stretched new codebook that has MSE=212.9.

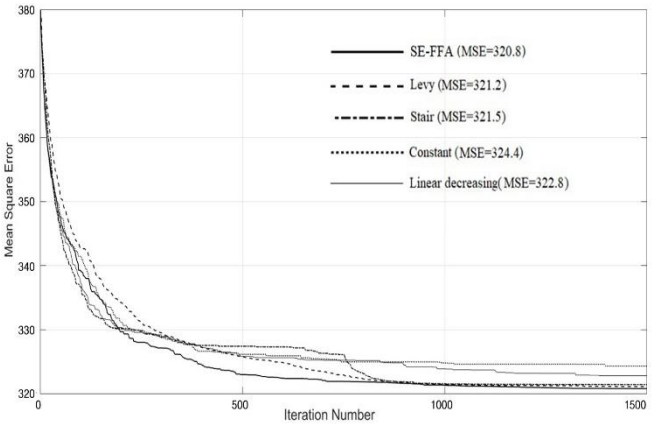


Figure 7. Original 256×256 pixel size of high contrast *lena* image with COV=0.530. SE-FFA convergence results with the compression ratio of 0.122 bpp and performance comparison of different radius types starting from stretched new codebook that has MSE=379.9.

It is seen that the high contrast images need much more iteration number for codebook convergence. On the contrary, for the low contrast ones the slope of exponential radius function may be more steep. In this work a formulation is defined by the analysis of COV versus a constant. In order to find the relationship between the COV and a constant in Table 1, a curve fitting toolbox of MATLAB is used. The Root Mean Square Error (RMSE) values of the fitting functions are compared. The benchmark images of high contrast lena, peppers, cameraman chemical plant, normal contrast images of aerial, boat and low contrast images of airplane and moon are used for finding the relationship between COV and a. The Gaussian function, which perfectly represents the (COV, a) data with the minimum RMSE value, is seen in Figure 5.

Table 1. COV versus a constant curve fitting analysis.

Function	Equation	Constants	RMSE
Exponential	$y=m.e^{n.x}$	m=0.6445, n=-266	0.03307
Fourier	$y = a_0 + a_1 cos(wx) + b_1 sin(wx)$	a_0=0.3151 a_1=0.2418 b_1=-0.06671 w=460.8	0.01979
Gaussian	$y=a_1.e^{-\left(\frac{x-b1}{c1}\right)^2}$	a_1=0.5538 b_1=-0.0002681 c_1=0.004101	0.01796
Linear Fitting	$y = a_0 sin(x-\pi) + a_1(x-10)^2 - c$	a_0=1.096e+05 a_1=-5474 c=5.474e+05	0.01822
Polynomial	$y=a_0.x+a_1$	a_0=-101.9 a_1=0.6007	0.01852
Power	$y=m.x^n$	m=0.03346 n=-0.3771	0.06553
Custom Equation	$y=m.e^{-nx}+p$	m=238 n=0.4304 p=-237.4	0.02031
Sum of Sine	$y=a_1 sin(b_1 x+c_1)$	a_1=0.7016 b_1=174.3 c_1=2.177	0.01804

4. Experimental Results

4.1. Variable Diameter Performance Comparison

In this chapter starting from a statistical stretched initial codebook, the performance of the proposed SE-FFA is compared by the different radius types such as constant, exponential, levy, stair and linear decreasing. Three types of images are used for the test. Low contrast moon image with COV=0.217, high contrast lena with COV=0.530 and normal contrast boat that has COV=0.35. It is observed that smart exponential diameter selection has a superiority to the other variants of radius types. All tests begin at radius=2 pixel, total iteration number is 1500 and the radius functions are given in Table 2. Convergence graphics of SE-FFA are shown in Figures 6, and 7 according to different radius types and different standard images of high, low and normal contrast images. These standard images have 256 gray levels and 256×256 pixels size. The codebook size is selected as 8 which means compression ratio is 0.122 bpp.

4.2. Standard FFA and SE-FFA Comparison

The reconstructed image of standard FFA and new introduced SE-FFA algorithms are compared according to MSE criteria at the same compression rate. In the tests high contrast lena image and low contrast moon images are used. Since the FFA algorithm uses a random initial

codebook and constant radius, it rarely reaches the global solution. On the contrary, our proposed algorithm SE-FFA starts from a robust stretched codebook and uses smart exponential radius for the fruit flies. Therefore, our algorithm is superior to the standard one according to MSE criteria and smart fruit flies reach the global minimum. Here, a high contrast image of lena with COV=0.530 and low contrast image of moon with COV=0.217 are used. Firstly, in these tests our proposed algorithm SE-FFA starts by using a robust initial codebook that stretches the gray level scale, where the standard FFA technique begins from a randomly selected codebook. Secondly, the proposed SE-FFA technique uses a smart exponential function radius based route based on the COV value of the test image. On the contrary, the standard FFA uses fixed diameter strategy. These comparisons for standard Lena image are seen in Figures 8, and 9. Here the standard FFA algorithm uses fixed 2 pixels wide search radius where the SE-FFA technique uses a smart exponential decreasing radius from 2.0 to 0.0 pixel size.

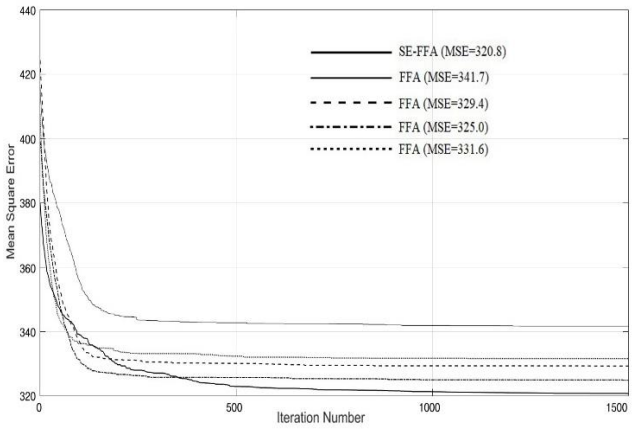


Figure 8. Original 256x256 pixel size of high contrast *lena* image with COV=0.530. SE-FFA convergence results with the compression ratio of 0.122 bpp, and final codebook mean square error of MSE=320.8. Standard FFA at the same compression ratio, fixed diameter search strategy and the resulting codebook errors of FFA changes from MSE=331.6 to MSE=341.7.

a) Original *Lena* image. b) SE-FFA MSE=320.8.

c) FFA MSE=325.0. d) FFA MSE=341.7.

Figure 9. a) Original 256x256 pixel size of high contrast *Lena* image with COV=0.530. b) Proposed SE-FFA with MSE=320.8. c) Classical FFA result with MSE=325.0 d) Classical FFA result with MSE=341.7 the radius is selected as 2 pixels, compression ratio is selected as 0.122 bpp.

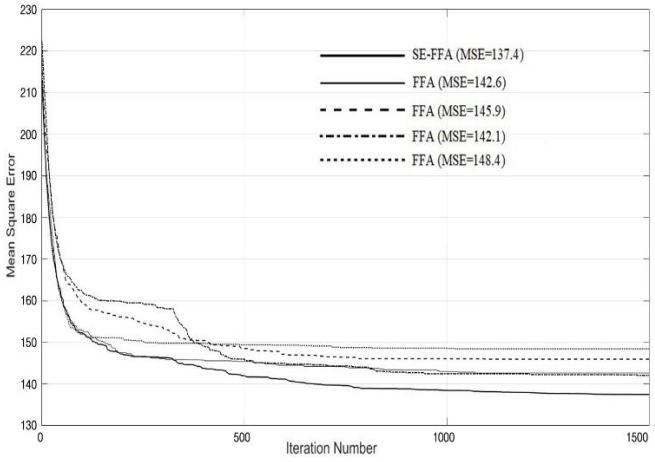
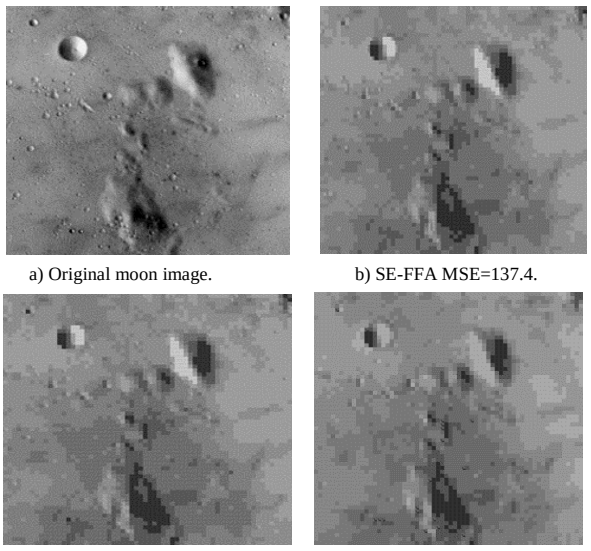


Figure 10. Original 256x256 pixel size of low contrast *Moon* image with COV=0.217. SE-FFA convergence results with the compression ratio of 0.122 bpp and final codebook error of MSE=137.4. Standard FFA at the same compression ratio, fixed diameter search strategy and the resulting codebook errors of FFA changes from MSE=142.6 to MSE=148.4.



a) Original moon image. b) SE-FFA MSE=137.4.

c) FFA MSE=142.1. d) FFA MSE=148.4.

Figure 11. a) Original 256x256 pixel size of low contrast *Moon* image with COV=0.217. b) Proposed SE-FFA with MSE=137.4. c) Classical FFA with MSE=142.1 d) Classical FFA with MSE=148.4. The radius is selected as 2 pixels, compression ratio is selected as 0.122 bpp.

Table 2. Comparison of convolutional FFAs and SE-FFA by using MSE criteria.

Compression Ratio: 0.122bpp	COV	FFA	Linear Decrease FFA	Stair FFA	Levy FFA	SE-FFA
Peppers	0.447	335.8	326.5	323.4	329.1	320.2
Clock	0.308	306.3	299.9	298.7	301.4	295.1
Airplane	0.170	126.9	124.0	122.2	124.2	120.3
Boat	0.357	373.3	366.1	366.9	363.4	363.2
Chemical plant	0.405	359.1	357.5	356.3	358.0	355.9
Cameraman	0.530	411.3	407.1	409.5	410.1	405.4
Barbara	0.454	348.7	343.4	340.2	345.0	338.4
Einstein	0.344	269.3	235.9	238.3	246.1	229.2
Couple	0.313	309.7	304.8	305.0	307.6	303.0
Lena	0.530	324.4	322.8	321.5	321.2	320.8
Moon	0.217	145.6	138.6	137.8	138.9	137.4
Baboon	0.300	432.3	423.2	422.6	427.8	418.7
Aerial	0.322	528.5	524.6	521.1	525.7	518.4
Average	0.361	329.4	321.6	320.9	323.9	317.8

The convergences of the algorithms for the standard

MOON image are given in Figure 10. The visual comparisons of the proposed SE-FFA algorithm and the standard FFA are shown in Figure 11. It is seen that the proposed SE-FFA algorithm is better than the standard FFA one according to the MSE criteria. The compression ratio is set to 0.122. The new introduced technique reduces the error to MSE=320.8 for the standard 256x256 pixel sized LENA image. Similarly new SE-FFA reduces error to MSE=137.4 for the standard 256x256 pixel sized MOON image.

The compression ratio set to 0.122. As it is seen in Table 3 and Figure 8 to Figure 11, the proposed SE-FFA method is obviously produce more accurate results when compared with the other FFA types such as have linear decreasing radius, have stair type radius and have levy based radius. The MSE performance comparison of the standard FFA and different radius types of FFA such as linear decreasing, stair, Levy function and smart exponentional radius type are shown in Figure 12.

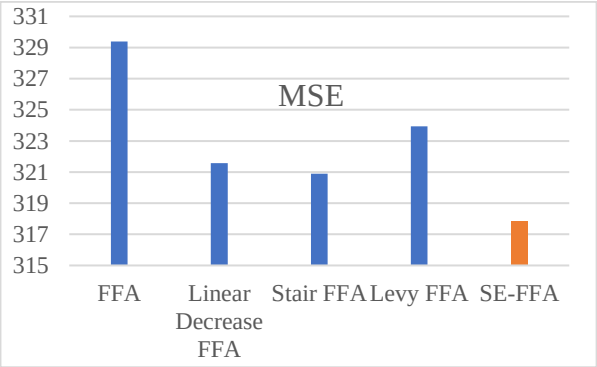


Figure 12. The average MSE performance results of the methods.

4.3. Results and Discussions

In this study, the FFA (Fruit Fly) algorithm is improved with an intelligent exponential flight to find one of the optimum codebooks. As the novel contribution, the disadvantage of the conventional FFA algorithm is overcome by using two new techniques.

The first one is a powerful codebook technique prepared for the FFA. This new source codebook is generated through statistical analysis to expand the color space of image blocks. Therefore, the fruit flies will be initialized properly and fly to the global solution without getting stuck at the minimum. The randomly initialized codebook of the FFA algorithm is changed by the new codebook that stretches the image block space. This is achieved by arranging the blocks according to their average. Then all the blocks are grouped. The number of group size is selected as same as the codeword number in the codebook.

The second improvement is about the search strategies of the fruit flies. A newly introduced smart exponential radius of the fruit flies makes the fruit flies' route robust and obtains the global optimum. Preprocessing first analyzes the input image to determine the standard deviation and coefficient of variation.

The coefficient of variation is used to determine if an image has low or high contrast. If the image has high contrast, it is difficult to process and additional iterative steps are required to find the codeword representing the image. On the other hand, when the contrast of an image is low, it is easier to imagine it with fewer codewords and fewer repetitive steps. Therefore, the slope of the exponential radius is determined by the coefficient of variation. If the image contrast is high, you will need to gradually decrease the slope of the exponential function one by one to give the fruit fly a chance, and increase the iteration steps to find the codewords needed to represent the image. Conversely, if the image has low contrast, it is selected so that the image is simple, and the slope of the exponential radius decreases sharply. This is because Drosophila does not go through too many repetitive steps to create codewords.

5. Conclusions

In this paper, the FFA is improved with a smart exponential flight method to construct a global optimum codebook. In this approach, if a fruit fly agent is captured by a local minimum point accidentally, smart exponential flight steps provide an opportunity to escape from it easily. Experimental studies have shown successful compression results. The numerical results show that the Drosophila Flight Based Algorithm (SE-FFA) using a smart index outperforms the various FFA search methods and the Fruitfly algorithm and provides a convenient and useful codebook.

References

[1] Chen P., Lin W., Huang T., and Pan W., "Using Fruit Fly Optimization Algorithm Optimized Grey Model Neural Network to Perform Satisfaction Analysis for E-Business Service" *Applied Mathematics and Information Sciences*, vol. 7, no. 21, pp. 459- 465, 2013. DOI:10.12785/amis/072L12

[2] Chiranjeevi K. and Jena U., "Image Compression Based on Vector Quantization Using Cuckoo Search Optimization Technique" *Ain Shams Engineering Journal*, vol. 9, no. 4, pp. 1417-1431, 2018. https://doi.org/10.1016/j.asej.2016.09.009

[3] Dai H., Zhao G., Lu J., and Dai S., "Comment and Improvement on A New Fruit Fly Optimization Algorithm: Taking the Financial Distress Model as an Example," *Knowledge-Based Systems*, vol. 59 pp. 159-160, 2014. https://doi.org/10.1016/j.knosys.2014.01.010

[4] Ding G., Dong F., and Zou H., "Fruit Flyoptimization Algorithm Based on A Hybrid

Adaptive Cooperative Learning and its Application in Multilevel Image Thresholding" *Applied Soft Computing*, vol. 84, 105704, 2019. https://doi.org/10.1016/j.asoc.2019.105704

[5] Feng H., Chen C., and Ye F., "Evolutionary Fuzzy Particle Swarm Optimization Vector Quantization Learning Scheme in Image Compression" *Expert Systems with Applications*, vol. 32, pp. 213-222, 2007. https://doi.org/10.1016/j.eswa.2005.11.012

[6] Fu Y., Zhou M., Guo X., and Qi L., "Stochastic Multi Objective Integrated Disassembly Reprocessing Reassembly Scheduling Via Fruit Fly Optimization Algorithm" *Journal of Cleaner Production*, vol. 278, pp. 123364, 2021. https://doi.org/10.1016/j.jclepro.2020.123364

[7] Horng M. and Jiang T., "Image Vector Quantization Algorithm via Honey Bee Mating Optimization" *Expert Systems with Applications* vol. 38, no. 3, pp. 1382-1392, 2011. https://doi.org/10.1016/j.eswa.2010.07.037

[8] Hu Y., Su B., and Tsou C., "Fast VQ Codebook Search Algorithm for Grayscale İmage Coding," *Image and Vision Computing*, vol. 26, no. 5, pp. 657-666, 2008. https://doi.org/10.1016/j.imavis.2007.08.001

[9] Huang H., Pan J., Lu Z., Sun S., and Hang H., "Vector Quantization Based on Genetic Simulated Annealing," *Signal Processing*, vol. 81, no. 7, pp. 1513-1523, 2001. https://doi.org/10.1016/S0165-1684(01)00048-2

[10] Ingaleshwar S., Dharwadkar N., and Jayadevappa D., "Water Chaotic Fruit Fly Optimization-Based Deep Convolutional Neural Network for İmage Watermarking Using Wavelet Transform," *Multimedia Tools and Applications*, pp. 1-25, 2021. DOI:10.1007/s11042-020-10498-0

[11] Jiang W., Wu X., Gong Y., Yu W., and Zhong X., "Holt-Winters Smoothing Enhanced By Fruit Fly Optimization Algorithm to Forecast Monthly Electricity Consumption" *Energy*, vol. 193, pp. 116779, 2020. https://doi.org/10.1016/j.energy.2019.116779

[12] Jindal R. and Singla S., "Latent Fingerprint Recognition using Hybrid Ant Colony Opt. and Cuckoo Search" *The International Arab Journal of Information Technology*, vol. 20, no. 1, pp. 19-28, 2023. https://doi.org/10.34028/iajit/20/1/3

[13] Kumar S., Fred A., Kumar H., Verghase P., and Daniel A., "Bat Optimization Based Vector Quantization Algorithm for Compression of CT Medical Images," *in Proceedings of the International Conference on Translational Medicine*, pp. 53-64, 2017. https://link.springer.com/chapter/10.1007/978-981-13-1477-3_5

[14] Li C., Xu S., Li W., and Hu L., "A Novel Modified Fly Optimization Algorithm for Designing The Self-Tuning Proportional İntegral Derivative Controller" *Journal of Convergence Information Technology*, vol. 7, no. 16, pp. 69-77, 2012. DOI:10.4156/jcit.vol7.issue16.9

[15] Li H., Guo S., Li C., and Sun J., "A Hybrid Annual Power Load Forecasting Model Based on Generalized Regression Neural Network with Fruit Fly Optimization Algorithm" *Knowledge Based Systems*, vol. 37, pp. 378-387, 2013. https://doi.org/10.1016/j.knosys.2012.08.015

[16] Li J., Pan, Q., and Mao K., "A Hybrid Fruit Fly Optimization Algorithm for The Realistic Hybrid Flowshop Rescheduling Problem in Steelmaking Systems," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 932-949, 2015. DOI: 10.1109/TASE.2015.2425404

[17] Lin S., " Analysis of Service Satisfaction in Web Auction Logistics Service Using A Combination of Fruit Fly Optimization Algorithm and General Regression Neural Network," *Neural Computational Applications*, vol. 22, pp. 783-791, 2013. DOI:10.1007/s00521-011-0769-1

[18] Linde Y., Buzo A., and Gray R., "An Algorithm for Vector Quantizer Design," *IEEE Transaction on Communications*, vol. 28, no. 1, pp. 84-95, 1980. DOI: 10.1109/TCOM.1980.1094577

[19] Lin Y. and Tai S., "A Fast Linde-Buzo-Gray Algorithm in İmage Vector Quantization," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 45, no. 3, pp. 432-435, 1998. DOI: 10.1109/82.664257

[20] Meng T. and Pan Q., "An İmproved Fruit Fly Timization Algorithm for Solving The Multi-Dimensional Knapsack Problem," *Applied Soft Computing*, vol. 50, pp. 79-93, 2017. https://doi.org/10.1016/j.asoc.2016.11.023

[21] Nasrabadi N. and King R., "Image Coding Using Vector Quantization: A Review," *IEEE Transactions on Communications*, vol. 36, no. 8, pp. 95-971, 1988. DOI: 10.1109/26.3776

[22] Pan J., "VQ Codebook Design Using Genetic Algorithms" *Electronics Letters*, vol. 31, no. 17, pp. 1418-1419, 1995.

[23] Pana S. and Chenga K., "An Evolution Based Tabu Search Approach to Codebook Design," *Pattern Recognition*, vol. 40, no. 2, pp. 476-491, 2007. https://doi.org/10.1016/j.patcog.2005.11.021

[24] Rani M., Rao G., and Rao B., "An Efficient Codebook Generation Using Firefly Algorithm for Optimum Medical Image Compression," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 4067-4079, 2020. DOI:10.1007/s12652-020-01782-w

[25] Sheng W. and Bao Y., "Fruit Fly Optimization Algorithm Based Fractional Order Fuzzy–Pid Controller for Electronic Throttle," *Nonlinear Dynamics*, vol. 73, no.1, pp. 611-619, 2013. https://doi.org/10.1007/s11071-013-0814-y

[26] Sun H., Lam K., Chung S., Dong W., Gu M., and Sun J., "Efficient Vector Quantization Using Genetic Algorithm," *Neural Computing and Applications*, vol. 14, pp. 203-211, 2005.

[27] Tsai C., Lee C., and Yang C., "A Fast VQ Codebookbook Generation Algorithm Via Pattern Reduction," *Pattern Recognition Letters*, vol. 30, pp. 653-660, 2009. DOI:10.1016/j.patrec.2009.02.003

[28] Tsai C., Tseng S., Yang C., and Chiang M., "PREACO: A Fast Ant Colony Optimization for Codebook Generation," *Applied Soft Computing*, vol. 13, no. 6, pp. 3008-3020, 2013. https://doi.org/10.1016/j.asoc.2013.01.017

[29] Wang L., Xiong Y., Li S., and Zeng Y., "New Fruit Fly Optimization Algorithm With Joint Search Strategies for Function Optimization Problems," *Knowledge-Based Systems*, vol. 176, pp. 77-96, 2019. https://doi.org/10.1016/j.knosys.2019.03.028

[30] Yang S., "Constrained-Storage Multistage Vector Quantization Based on Genetic Algorithm," *Pattern Recognition*, vol. 41, no. 2, pp. 689-700, 200. https://doi.org/10.1016/j.patcog.2007.05.011

[31] Yuan, X., Dai, X., Zhao, J., and He Q., "On A Novel Multi-Swarm Fruit Fly Optimization Algorithm and its Application," *Applied Mathematics and Computation*, vol. 233, pp. 260-271, 2014. https://doi.org/10.1016/j.amc.2014.02.005

[32] Zhang X., Xu, Y., Caiyang Yu, C., Heidari A., Li S., Chen H., and Li C., "Gaussian Mutational Chaotic Fruit Fly Built Optimization and Feature Selection," *Expert Systems with Applications*, vol. 141, pp. 112976, 2020. https://doi.org/10.1016/j.eswa.2019.112976

Ilker Kilic is working as an academical person at the Electrical and Electronics Engineering department of Pamukkale University. He had the PhD degree in 2003 on image&video processing from Dokuz Eylul University in TURKEY. His works focuses on image processing, transform techniques, image segmentation and bio-inspired optimization algorithms.