mathematics

Article

Performance Augmentation of Cuckoo Search Optimization Technique Using Vector Quantization in Image Compression

Aditya Bakshi [1], Akhil Gupta [2,*], Sudeep Tanwar [3,*], Gulshan Sharma [4], Pitshou N. Bokoro [4], Fayez Alqahtani [5], Amr Tolba [6] and Maria Simona Raboaca [7,8]

[1] Department of Computer Science & Engineering, Manipal Institute of Technology, Manipal 576104, Karnataka, India; aditya.bakshi@manipal.edu
[2] School of Electronics and Electrical Engineering, Lovely Professional University, Phagwara 144411, Punjab, India
[3] Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad 382481, Gujarat, India
[4] Department of Electrical Engineering Technology, University of Johannesburg, Johannesburg 2006, South Africa; gulshans@uj.ac.za (G.S.); pitshoub@uj.ac.za (P.N.B.)
[5] Software Engineering Department, College of Computer and Information Sciences, King Saud University, Riyadh 12372, Saudi Arabia; fhalqahtani@ksu.edu.sa
[6] Computer Science Department, Community College, King Saud University, Riyadh 11437, Saudi Arabia; atolba@ksu.edu.sa
[7] Doctoral School, University Politehnica of Bucharest, Splaiul Independentei Street, No. 313, 060042 Bucharest, Romania; simona.raboaca@icsi.ro
[8] National Research and Development Institute for Cryogenic and Isotopic Technologies—ICSI Rm. Vâlcea, Uzinei Street, No. 4, P.O. Box 7 Râureni, 240050 Râmnicu Vâlcea, Romania
* Correspondence: akhil.20239@lpu.co.in (A.G.); sudeep.tanwar@nirmauni.ac.in (S.T.)

Abstract: For constructing the best local codebook for image compression, there are many Vector Quantization (VQ) procedures, but the simplest VQ procedure is the Linde–Buzo–Gray (LBG) procedure. Techniques such as the Gaussian Dissemination Function (GDF) are used for the searching process in generating a global codebook for particle swarm optimization (PSO), Honeybee mating optimization (HBMO), and Firefly (FA) procedures. However, when particle velocity is very high, FA encounters a problem when brighter fireflies are trivial, and PSO suffers uncertainty in merging. A novel procedure, Cuckoo Search–Kekre Fast Codebook Generation (CS-KFCG), is proposed that enhances Cuckoo Search–Linde–Buzo–Gray (CS-LBG) codebook by implementing a Flight Dissemination Function (FDF), which produces more speed than other states of the art algorithms with appropriate mutation expectations for the overall codebook. Also, CS-KFGC has generated a high Peak Signal Noise Ratio (PSNR) in terms of high duration (time) and better acceptability rate.

Keywords: vector quantization (VQ); image compression (Img Comp); codebook; encoding

MSC: 62H35

check for updates

Citation: Bakshi, A.; Gupta, A.; Tanwar, S.; Sharma, G.; Bokoro, P.N.; Alqahtani, F.; Tolba, A.; Raboaca, M.S. Performance Augmentation of Cuckoo Search Optimization Technique Using Vector Quantization in Image Compression. *Mathematics* **2023**, *11*, 2364. https://doi.org/10.3390/math11102364

Academic Editor: Cheng-Chi Lee

Received: 25 April 2023
Revised: 10 May 2023
Accepted: 17 May 2023
Published: 19 May 2023

1. Introduction

In multimedia applications, Image Compression (Img Comp) plays a substantial role in establishing the reconstructed image with fewer bits. In the last ten years, interest in Img Comp has increased remarkably as many digital illustrations have been created with a few novel utilizations. Img Comp is vital in interactive media operations, especially cellular phones, web perusing, faxes, etc. The major limitations of Img Comp are the size and quantity degradation. The selected image in the system can be decoded and matched with the standard image using Img Comp. Lossy and Lossless compression techniques are two types of Img Comp techniques. In Lossless, the total number of bits in an image is decreased. The main advantage of Lossless compression is that an image's data remain intact while compressing the data. In Lossy, the Peak Signal Noise Ratio (PSNR) values are measured for lost data after compressing the image. The two major well-known standards of the Lossy compression scheme are JPEG and JPEG2000 [1]. There are two methods of Img Comp: frequency and spatial domain. The reduced (i.e., amount of bits) image is directly compared with the original image in the spatial domain. The frequency domain can be found using transformation, quantization, and encoding. In an uncorrelated domain, changing an image's pixels is called transformation. To attain the enormous compression ratio, quantization is applied to reduce the actual bits. Here, fewer data are taken from every pixel in an image. Scalar Quantization (SQ) and Vector Quantization (VQ) are two types of quantization. Using function (Q), the process of mapping the input value (x) onto the resultant value (y) is called Scalar Quantization. The Linde–Buzo–Gray (LBG) [2] algorithm is the prime example of optimal vector quantization with minimum distortion. In VQ, code words are represented with image vectors and can be shown in image blocks. The mechanism of generating the codebook in VQ is divided into three parts: (1) Design procedure of the codebook, (2) Encoding procedure of an image, and (3) Decoding procedure. Before encoding and decoding the image, the design procedure of the code word has been implemented. When the compression ratio is high, VQ yields good results. In VQ, vectors are coded using a transformed image and kept in the codebook using an index file. The bit rate of transmission is minimized using the required data index file for every block of pixels. To design the appropriate codebook used in image encoding and decoding procedures, the codebook design technique comes into the picture.

1.1. Literature Survey

For implementing the proposed work, a literature survey has been carried out in which Patane et al. proposed an updated LBG calculation that enhances the neighborhood's ideal arrangement of a LBG calculation [3]. Wang et al. proposed a novel Quantum Swarm Developmental calculation (QSE). They showed the comparison view of the quantum-motivated transformative calculation (QEA) [4]. Karaboga et al. proposed an ABC algorithm for calculating multivariable capacities with enhanced capabilities [5]. Jiang et al. proposed a point-by-point portrayal of the HBMO (Honeybee mating advancement) calculation and improved the implementation of the LBG method using the VQ codebook [6]. Horng et al. proposed another calculation termed BA-LBG, which utilized BA on starting arrangement of LBG [7]. Karri et al. presented a Cuckoo Search (CS) meta-examining advancement calculation with the LBG codebook by implementing and calculating Mantegna's calculation using Gaussian Dissemination [2]. Yang et al. displayed a broader correlation examination utilizing some standard test capacities and de-marked test functions using the stochastic method [8]. Yang et al. planned to figure out another meta-examining calculation, termed Cuckoo Search (CS), considering investigating revising difficulties [9]. Zhou et al. proposed a model for estimating the sparsity of an image in which the offline training phase and online estimation phase have been applied to image patches. The sparsity range is determined using offline training, and a group of Pareto solutions is obtained using MOEA/D for the training patch. Secondly, using online estimation, the most similar training patch is selected using the sparsity range. A sparsity-restricted greedy algorithm (SRGA) can be obtained using a sparse representation vector within that range [10]. Zhou et al. proposed a two-phase algorithm for sparse signal reconstruction. A group of robust solutions is generated using the decomposition-based multi-objective evolutionary algorithm in phase 1. This can be achieved by optimizing the norm of the solutions. The statistical features concerning each entry among these solutions are extracted to remove the noise interruption. Locating the nonzero entries, a forward-based selection method is proposed for updating the set more precisely for these features in phase 2 [11]. Yan Qiu et al. proposed a novel objective function by combining local features with classifiers such as region purity and region-based distance metrics. The authors proposed a purity-based LFS (RP-LFS) where the proportion of the selected features and region-based distance metric is calculated. An improved non-dominated sorting genetic algorithm III is also presented with the RP-LFS method. To improve the ability to search for better solutions, a network-inspired crossover operator and a quick bit mutation are applied [12]. Wenjun Zhang et al. presented a problem-specific non-dominated sorting genetic algorithm (PS-NSGA) by minimizing three objectives of FS. In PS-NSGA, the preferred domination operator achieves higher classification accuracy in the population. High efficiency is achieved using the bit mutation method that overcomes the limitation of traditional bit string mutation [13]. Genetic algorithms play a significant role in optimization problems as they are very different from random algorithms. They generally belong to the class of probabilistic algorithms and usually require directed and stochastic search. All other methods process a single point of the search space, but genetic algorithms maintain a population of potential solutions. Genetic algorithms are gradient-free, parallel optimization algorithms that use a performance criterion for evaluation. This is why genetic algorithms are more robust than existing directed search methods.

1.2. Organization

The paper's organization is as follows: Section 1 introduces image compression. Here we discuss the literature survey of different optimization problems and their solutions. In Section 2, we discuss the codebook design methods for VQ. In Section 3, we discuss the proposed CS-KFGC Vector Quantization method along with its algorithm. In Section 4, the simulation results of the proposed work are shown, and finally, Section 5 concludes the paper.

2. Codebook Design Methods for VQ

The VQ is one way that summarizes Img Comp Codebook layout is very important as it is essential in designing a VQ that diminishes the deformity between recreated and authentic images within a restricted computing period. Figure 1 presents the ciphering (encoding) and the VQ's deciphering (decoding) mechanism. Here, the different working principle has been adopted using three different slabs. Slab 1 is an encoding segment that generates indexing, codebook generation, and image vectors. Generating the image vector divides the input image into immediate and non-overlapping blocks. VQ's major task is the generation of an efficient codebook. The mechanism for grouping equal-size codebooks with non-overlapping block sizes is called codebook. The algorithm that generates an efficient codebook is considered successful. With the help of the index table, indices are properly indexed with each vector after successfully generating the codebook. In the end, indices are communicated to the receiver. The channel required to communicate the indexed numbers to the receiver is shown in slab 2. Slab 3 is a decoding section comprising a codebook, index table, and the reconstructed image. The receiver index table decodes the received indexed numbers. Lastly, the receiver codebook is matched with the transmitter codebook. The image (dimension $(\mathrm{M} \times \mathrm{M})$ elected VQ is separated within $\mathrm{M_b}$ $(\frac{\mathrm{M}}{\mathrm{m}} * \frac{\mathrm{M}}{\mathrm{m}})$ blocks among dimensions m × m pixels. The separated image section with other words training vectors of dimension m × m picture element (pixel) can be expressed as:

$$\mathrm{Z_j}\ (\mathrm{j} = 1,\ 2,\ 3,\ \ldots,\ \mathrm{M_b})$$

The codebook contains an array of code words, $\mathrm{C_j}$ (where j = 1 … $\mathrm{M_b}$) is the $\mathrm{j^{th}}$ code word. The comprised code words in Codebook is $\mathrm{M_b}$.

Individual separated image vectors are proximate by the tabulated form of code words, hinged on the minimal Euclidean distance with conforming codewords and vectors. To estimate the distances between image vectors and codewords, the Euclidean distance is required in VQ [14]. If p is an image vector and $\mathrm{q_i}$ is a codeword then their Euclidean distance is calculated as follows:

$$\mathrm{d}(\mathrm{p},\ \mathrm{q_i}) = \|\mathrm{p} - \mathrm{q_i}\|^{\,2} = \textstyle\sum_{\mathrm{n}=1}^{\mathrm{m}} (\mathrm{p_n} - \mathrm{q_{in}})^2 \quad (1)$$

Here, p_n and q_{in} denote the nth components of p and q_i, respectively, and m denotes the dimensionality of p and q_i. From this equation, the Euclidean distance needs m subtractions and m multiplications for the calculation of $m(p_n - q_{in})^2$s.

The enciphered outcomes are given as an index table.

The deformity (DF) with training vectors, along with the codebook, is given as follows:

$$DF = \frac{1}{M_c} \sum_{i=1}^{M_c} \sum_{j=1}^{M_b} u_{ji} \cdot \|Z_j - C_i\|^2 \quad (2)$$

$$DF = \sum\nolimits_{i=1}^{N_c} u_{ji} = 1\, j = \{1, 2, \ldots, M_b\} \quad (3)$$

where, u_{ji} =1 if Z_j is in the j^{th} cluster, otherwise it will be zero.

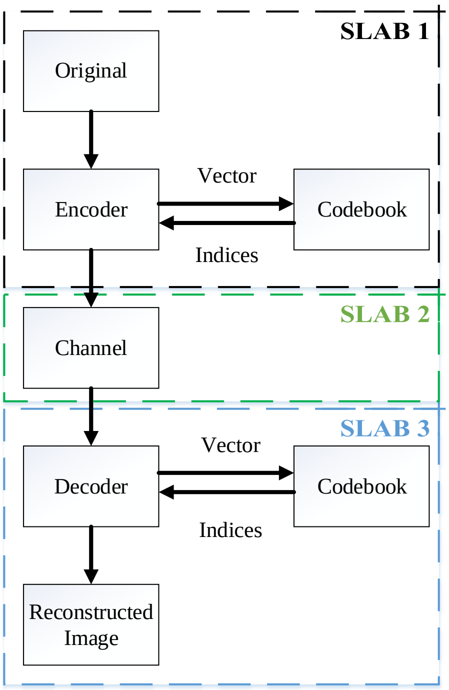


Figure 1. Process of practice of VQ.

In the next section, detailed mathematical explanations of all the VQ algorithms are given.

(a) Linde–Buzo–Gray VQ Technique (LBG)

The Linde–Buzo–Gray (LBG) algorithm [2] is a primitive algorithm and the most-used VQ procedure. It is understandable, versatile, and flexible, but the results need to improve for regional optimization. LBG algorithm calculations depend on minimal Euclidean distances with an image vector and a corresponding code word. The major limitation of the LBG algorithm is that its codeword length is fixed, and it fails to give good results in a global optimum codebook and gets stuck in a local optimum. The ultimate additional practiced procedure in VQ is termed as Generalized Lloyd Algorithm (GLA). The steps for implementing the LBG algorithm are as follows:

Step 1: Initiate the basic codebook C_1 of dimension M. Enable repetition counter as $n = 1$ and $D_1 = \infty$, i.e., distortion.

Step 2: Using $C_n = \{Y_j\}$ as a codebook, segregate the training set within array sets S_j applying the prerequisite of the technique named as adjoining neighbor.

Step 3: Assign the input vectors to the introductory code vectors. After that, the centroid of the portioned region has been calculated using step 2. The improved codebook can be defined as $C_n + 1$.

Step 4: Average distortion is calculated in $D_n + 1$. Whenever $D_n - D_n + 1 < T$ stop, otherwise increase $n = n + 1$ and rerun step 2–4. The deformity becomes small after repeatedly carrying out the LBG algorithm.

(b) Particle Swarm Optimization–Linde–Buzo–Gray VQ Technique (PSO-LBG)

PSO [2] can be described as the collective activities of a group of birds or the schooling of fishes. Earlier, many researchers used the PSO technique in their work as it gives optimized results. Junhao Kang et al. proposed a discretization-based multi-objective optimization framework for FS. The authors proposed a flexible cut-point PSO (FCPSO) model with relevant features to select an arbitrary number of cut points for discretization. In FCPSO, a novel adaptive mutation operator effectively finds the relevant features and eliminates the duplicate features [15]. Yu Zhou et al. proposed a two-level particle cooperation strategy using a binary particle swarm optimization that reformulates the FS optimization problem, comprising three objectives to be minimized. In the first level, randomly generated ordinary particles and strict particles filtered by ReliefF are combined to maintain rapid convergence. In the second level, cooperation between particles is conducted using a decomposition multi-objective optimization framework to process the search for Pareto solutions more efficiently [16]. An adequacy evaluation function also consigns an adequacy valuation of a particle's locus. Each fragment records a pair of loci.

The PSO models are defined as in two types:

- h_{best} model
- q_{best} model.

The PSO h_{best} reproduction presented [17] modifies a VQ by inserting the outcome of a LBG algorithm h_{best} fragment. Currently, PSO codebooks amend their significance established on their preceding training and the leading training of the swarm. Here, fragments have been denoted by codebook. The major limitation of the PSO-LBG technique is that convergence speed of PSO-LBG is very slow and the space for searching is very large.

The step for performing PSO-LBG technique are as follows:

Step 1: Compute the LBG algorithm and the overall best codebook (h_{best}) is applied.

Step 2: Load in the other codebooks with irregular amounts and their analogous momentum.

Step 3: Find number of Fitness values by Equation (4) for each codebook:

$$\text{Fitness}\ (C) = \frac{1}{DF} = \frac{M_v}{\sum_{i=1}^{M_c} \sum_{j=1}^{M_b} u_{ji} \cdot \|Z_j - C_j\|^2} \tag{4}$$

Step 4: If the new adequate value is better than the past adequate (q_{best}) then define its conforming fresh fitness as q_{best}.

Step 5: Select the topmost adequate value from all the codebooks. If it is superior to h_{best}, then put in place of h_{best} as best adequate amount.

Step 6: Change the rates using Equation (5). After that revise a fresh locus by Equation (6) for each fragment and return to step 3:

$$v_{jL}^{m+1} = v_{jL}^{m} + a_1 s_1^m (\, {q_{bestik}}^m - Z_{ik}^m + a_2 s_2^m \left({h_{bestk}}^m - Z_{jL}^n \right) \tag{5}$$

$$Z_{jL}^{m+1} = Z_{jL}^{m} + v_{jL}^{m+1} \tag{6}$$

where L = the count of results, j = the locus of the fragment, a_1 and a_2 are justifiable and social research estimates, respectively. s_1 and s_2 are arbitrary counts.

Step 7: Just before a halting benchmark is encountered (Maximum reiteration), re-do steps 3–7.

(c) Quantum Particle Swarm Optimization–Linde–Buzo–Gray VQ Technique (QPSO-LBG)

The QPSO [2] is different from PSO as it calculates the local particle q_i using q_{best} and h_{best} as shown in Equation (7):

$$q_i = s_1\, q_{best^i} + \frac{s_2\, h_{besti}}{s_1 + s_2} \tag{7}$$

Additionally, two constants, e and f, are introduced to restore the locality of fragments linked with the regional particle. To condense, triple constants are used for restoring the particle [18,19]. The slow convergence speed is the major drawback of this QPSO-LBG algorithm. The steps for implementing the algorithm are presented:

Step 1: LBG algorithm may be run first.

Step 2: Load the result of LBG algorithm to one fragment and compute locus of remaining fragments. Check speed of all fragments in different conditions.

Step 3: Estimate the fitness rate for each fragment using Equation (4).

Step 4: Differentiate each individual fragment's fitness rate with previous fragment's particular best value. Update h_{best} and stop recording the ongoing locus as the fragment's particular best locus.

Step 5: Discover the utmost fitness rate of all fragments. If the rate is superior to q_{best}, replace g_{best} with the fitness rate and take proceedings of the overall best locus.

Step 6: However, include the constants s_1, s_2 and u whose rate ranges from zero to one, and the local point q_i established in Equation (7).

Step 7: Reconsider the locus of fragment Z_j as given by:

$$k_i = x\left|Z_j - q_i\right| \tag{8}$$

If e > 0.5,

$$Z_j(t+1) = q_i - k_i \cdot \ln\left(\frac{1}{e}\right) \tag{9}$$

otherwise,

$$Z_j(t+1) = q_i + k_i \cdot \ln\left(\frac{1}{e}\right) \tag{10}$$

where x is non-pessimistic stable and is below $1 = \log\sqrt{2}$ and t is the present reiterative period.

Step 8: Return steps 3 to 8 just before stop record is executed.

(d) Honeybee Mating-LBG VQ Technique (HBMO-LBG)

A honeybee group characteristically comprises a single egg-laying long-lived queen, anywhere from zero to more than a few thousand drones, and customarily 10,000–60,000 laborers [2]. A reproducing journey initiates with a dance by the queen that initiates a reproducing journey for the duration; the drones follow the queen and breed with her in the air. In each reproduction, semen enters the spermatheca and merges to fertilize the ova. Whenever a queen lays fertilized eggs, it uses a concoction of the semen gathered in the spermatheca to inseminate the oospore. In processes, the reproducing flight is reflected as a set of evolutions in a state-zone where the queen travels amid the peculiar states in a few velocities and breeds with the drone confronted at individual case expectations [20,21]. Additionally, the queen visualizes with specific power content for the duration of the flight reproducing and returns to its nest when the power is inside, beginning from an empty to full spermatheca [22]. For implementing the HBMO algorithm, large numbers of parameters are required, and its computation time is higher than other algorithms.

The step for implementing the HBMO-LBG algorithm is following:

Step 1: (Accomplish the drone array and original queen)

In this type, the outcome of the LBG algorithm is an antecedent array to the queen P, and then the cluster of drones (minor outcomes), $\{DR_1, DR_2, \ldots DR_n\}$ are engendered under any circumstance. The adequacy of each individual drone is estimated and the best one is chosen, importing as the DR_{best}. On the assumption of the adequacy for DR_{best} is best than the adequateness of the P, the queen P is put in place by $\mathrm{DR_{best}}$. The $\mathrm{p_j}$ is $\mathrm{j^{th}}$ codeword of the codebook of queen P, and the $\mathrm{p_j^h}$ is the $\mathrm{h^{th}}$ aspect of the $\mathrm{j^{th}}$ code word.

Step 2: (Reproducing flight)

In this period within the structure, the queen establishes the fitness reproducing array starting with the set of drones conforming to Equation (11).

$$\mathrm{Q(P, DR)} = \exp[\frac{-\Delta(\mathrm{f})}{\mathrm{R(t)}}] \tag{11}$$

The queen's speed is reduced before the individual reproducing procedure just before its speed is beneath the finest tiniest hovering speed. Let semen be a group of the sperms conferred in the abide by equations:

$$\mathrm{P} = (\mathrm{p_1, p_2, p_3 \ldots .. p_k}), 1 \ \leq \mathrm{k} \ \leq \mathrm{M_c} \tag{12}$$

$$\mathrm{p_j} = \left(\mathrm{p_j^1, p_j^2, p_j^3 \ldots .., p_j^L}\right) \tag{13}$$

where the $\mathrm{R_{qi}}$ = $\mathrm{i^{th}}$ semen in the spermathecal and the $\mathrm{R_{qi}}$, k = $\mathrm{k^{th}}$ code word of the codebook conferred by the $\mathrm{R_{qi}}$.

Step 3: The best progeny, progeny best with utmost adequacy rate is chosen from all progeny as the original queen.

Step 4: If the adequacy of progeny best is superior to the queen, we replace the queen by progeny best.

Step 5: Check the ending benchmark, assuming the completion benchmark is satisfied then finish the algorithm, otherwise discard all preceding trial results (progeny set). Then repeat step 2 until the designated reiteration is concluded.

(e) Firefly Algorithm-LBG VQ Technique (FA-LBG)

FA was created by Yang [8,9]. In FA, characteristics of fireflies and flashing pattern are required for illumination and, using flickering fireflies, the inferior brightness will proceed towards the superior brightness. On the assumption that there is no brightness other than a specific firefly, it will proceed under different circumstances. A firefly chooses any periodically diminishing function of distance $\mathrm{s_{i,j}} = \mathrm{df}(\mathrm{z_j},\ \mathrm{z_i})$ to the particular jth firefly, e.g., the exponential function:

$$\mathrm{s_{i,j}} = \|\mathrm{z_i} - \mathrm{z_j}\| \tag{14}$$

$$\beta \leftarrow \beta_o e^{-\mu s_{i,j}} \tag{15}$$

where, β_o = allure at $\mathrm{s_{i,j}}$ = 0, μ = light retention belonging at the origin. The motion of a firefly i is fascinating to another additional firefly j is driven as:

$$\mathrm{z_{i,l}} \leftarrow (1-\beta)\mathrm{z_{i,l}} + \ \beta \mathrm{z_{j,l}} + \mathrm{u_{i,l}} \tag{16}$$

$$\mathrm{u_{i,l}} = \alpha(\mathrm{rand1} - \frac{1}{2}) \tag{17}$$

The specific firefly $\mathrm{z_{i,}}$ with maximum adequacy will proceed under any circumstances according to equation:

$$\mathrm{z_{i,l}^{max}} \leftarrow \mathrm{z_{i,l}^{max}} + \ \mathrm{u_{i,l}^{max}} \text{ for } \mathrm{l} = 1, 2, 3 \ldots, \mathrm{c}. \tag{18}$$

$$\mathrm{u_{i,l}^{max}} = \alpha(\mathrm{rand2} - \frac{1}{2}). \tag{19}$$

when rand1 $\approx$ U (0, 1) rand2 $\approx$ U (0, 1) = arbitrary count regained from the homogeneous dissemination. The illumination of a firefly is overwhelmed or derived by the prospect of the adequacy function. For improvement complication, the illumination IM of a firefly at a specific locus z can be defined as IM(z), that is, equivalent to the rate of the adequacy function. The details of the FA-LBG algorithm are given by:

Step 1: (Achieve the introductory results and inured limits)

Here, the LBG algorithm codebook is empowered with one introductory result, and then the remaining introductory results, z_i, (i = 1, 2, . . . , n − 1) are engendered under any circumstances. Each result is the codebook with no code word.

Step 2: (Prefer the prevailing finest result)

Step 2 prefers the finest result from all results and is exemplified as z_{max}, i.e.:

$$i^{max} = \mathrm{argmax}_i \text{ Adequacy } (z_i) \tag{20}$$

$$z_i^{max} = \mathrm{argmax}_{xi} \text{ Adequacy } (z_i) \tag{21}$$

Step 3: (The motion of a firefly z_j is to another extra-alluring firefly z_i)

In step 3, each result z_j computes its adequacy valuation as conforming the illumination of the firefly. Because every result is z_j, this step unsystematically chooses another result z_j with additional brightness and therefore migrates to it, derives nearness by statements:

$$s_{ij} = \|z_i - z_j\| = \sqrt{\sum_{l=1}^{M_c} \sum_{h=1}^{L} (z_{i,l}^h - z_{j,l}^h)^2} \tag{22}$$

$$\beta = \beta_0 e^{-\mu s_{ij}} \tag{23}$$

$$z_{j,l}^h = (1-\beta) z_{i,l}^h + \beta z_{j,l}^h + u_{j,l}^h \tag{24}$$

where $l = 1, 2, \ldots . M_c$, $h = 1, 2, \ldots, L$, $u_{j,k}^h$. u is an unsystematic number ranging from zero to one.

Step 4: (Analyze the completion benchmark)

The algorithm is resolved if input (round number) is l and the preferred result maximum i is the output. Therefore, l represents the best result, i.e., $z_i{}^{max}$. The best result $z_i{}^{max}$ with its locus shown as:

$$z_{i^{max},l}^h \leftarrow z_{i^{max},l}^h + u_{i^{max},l}^h \tag{25}$$

where $l = 1, 2, 3 \ldots ., M_c$, $h = 1, \ldots,$ $u_{i^{max},l}^h$ is an unsystematic number ranging from zero to one. The authors used this algorithm and correlated it with the outdated LBG, PSO-LBG and QPSO-LBG and found that FA-LBG requires very minimum computation time, and the total count of limits is also minimum as compared to HBMO-LBG.

(f) Cuckoo Search-LBG VQ Technique (CS-LBG)

PSO generates an efficient codebook, but convergence has a huge instability whenever particle velocity is high [2]. FA gives a global codebook, but when the search space is brighter, it experiences a problem. The Cuckoo Search algorithm is an optimization algorithm (based on breeding and behavior method of cuckoo birds) and using one tuning parameter. a global codebook is generated [23]. The Cuckoo Search algorithm applies to two types of problems, i.e., linear and nonlinear [24]. Its reproduction approach is the most important part that inspires researchers, as cuckoo birds have emitted beautiful sounds. The eggs of cuckoos are laid in the nests of host birds [25]. The host birds will eject the eggs if they recognize them as ones they did not lay. Different parasitic species can lay colored eggs that match the eggs of passerine hosts, but in contrast, non-parasitic cuckoos lay white eggs [26]. In some cases, the egg color and its patterns can be mimicked by female cuckoos in selecting a host nest. This increases productivity for cuckoos as host birds throw away fewer eggs from the nest. The CS-LBG gives good results as it solves the problem of slower convergence with fewer computation parameters. Further modifications can be made in terms of slower convergence speed which is explained in the proposed algorithms.

Furthermore, incubation periods for birds are short; therefore, non-parasitic cuckoos can fly, leaving their nest [27]. The cuckoo's current position is centered on the cuckoo breeding process, and the number of random steps is chosen with a random walk. For a proper breeding process, exploration, exploitation, intensification, and diversification play an essential role in random walks [2]. However, a probability distribution function such as Gaussian, normal, and Lévy distributions [2] is required for random walk and step size forging. The Lévy distribution function follows Lévy flight by a random walk. Mantegna's algorithm gives both positives and negatives, and the uniform distribution function follows the direction of the walk. The Lévy distribution function as follows:

$$D(r, \mu, \gamma) = \left\{ \begin{array}{c} \sqrt{\frac{\mu}{2\Pi}} \exp\left[-\frac{\gamma}{2(r-\gamma)}\right] \frac{1}{(r-\gamma)^{\frac{3}{2}}} \quad 0 < \gamma < r < \infty \\ 0 \quad \text{alternatively} \end{array} \right\} \tag{26}$$

where $\gamma > 0$ is the smallest step. If $r \to \infty$, then the above-mentioned equation will get transformed:

$$D(r, \mu, \gamma) \approx \sqrt{\frac{\mu}{2\pi}} \frac{1}{(r)^{3/2}} \tag{27}$$

The size of an arbitrary cuckoo step is described by the mathematical statement of the Mantegna algorithm, i.e., random steps = $\frac{\gamma}{\upsilon^{1/\beta}}$ where γ and υ are Gaussian disseminations shown as:

$$L(r) = \frac{1}{\pi} \int_0^\infty \cos(\tau r) e^{-\tau \xi^\beta} d\tau \text{ with } \beta = 2 \tag{28}$$

From the above equation:

$$\gamma \approx N(0, \varphi_\gamma^2)\ \upsilon\ \approx N(0, \varphi_\upsilon^2)$$

where N() is the normal dissemination function in the mathematical statement:

$$N(\gamma, \varphi^2) = \frac{1}{\sqrt[\varphi]{2\pi}} \exp\left[-\frac{(z-\gamma)^2}{2\gamma^2}\right] \quad -\infty < z < \infty \tag{29}$$

$$\varphi_\gamma = \left\{ \frac{\Gamma'(1+\beta)\sin(\frac{\Pi\beta}{2})}{\Gamma'\left[\frac{1+\beta}{2}\right] \beta 2^{(\beta-1)/2}} \right\}^{\frac{1}{\beta}} \text{ and } \varphi_\upsilon = 1 \tag{30}$$

where gamma function (Γ') is given as follows:

$$\Gamma'(\beta) = \int_0^\infty u^{-t} t^{\beta-1} dt \tag{31}$$

The steps for implementing the (CS-LBG) algorithm are as follows:

Step 1: (Solutions and Parameters initialization): Initialize each nest containing a single egg, a mutation probability (Q) and a tolerance with the number of host nests. Run the rest of the nests randomly but the LBG algorithm is assigned as one of the nest/eggs.

Step 2: (Best solution selection): Current best nest nest$_{\text{best}}$ is selected after calculating the fitness of all nests.

Step 3: (Use of the Mantegna algorithm for new solution generation): Using random walk, novel nests (Nest$_{\text{novel}}$) are generated that are current best nest. This arbitrary walk follows the Lévy dissemination function and is followed by random walk that obeys Mantegna's algorithm. A novel nest is given as:

$$\text{Nest}_{\text{novel}} = \text{Nest}_{\text{original}} + \xi \otimes \text{Levy}(Z)$$

where, ξ = point size usually equal to 1, Levy(Z) is Levy Dissemination function obtained from the equation Step of arbitrary walk = $\frac{\gamma}{v^{1/\beta}}$ and $\varphi_{\gamma} = \left\{ \frac{\Gamma'(1+\beta)\sin(\frac{\Pi\beta}{2})}{\Gamma'\left[\frac{1+\beta}{2}\right]\beta 2^{(\beta-1)/2}} \right\}^{\frac{1}{\beta}}$ and $\varphi_{\upsilon} = 1$.

The above equation can be written as:

$$\mathrm{Nest}_{\mathrm{novel}} = \mathrm{Nest}_{\mathrm{original}} + \ \mathrm{step} \ \times (\mathrm{Nest}_{\mathrm{best}} - \ \mathrm{nest})$$

Step 4: (Substitute novel nests by discarding worst nest): If mutation expectation (Q) is smaller than generated random number (L) then the novel nest has been replaced by worst nest, keeping the finest nest unaffected. The novel nest is shown as:

$$\mathrm{Nest}_{\mathrm{novel}} = \mathrm{Nest}_{\mathrm{original}} + (\mathrm{L} \times \ \mathrm{stepsize})$$

where Step size $=$ s $\times$ nest$_{\mathrm{rand}}$ – nest$_{\mathrm{rand}}$, r is a random number.

Step 5: Select best nest using fitness function after ranking of all nests.

Step 6: Before finishing, repeat step 2 to step 4.

It was observed by the authors that the PSNR and its characteristic of the reconstructed image obtained using the CS algorithm generated good results as compared to LBG, PSO-LBG, QPSO-LBG, HBMO-LBG, and FA-LBG [28]. The CS-LBG algorithm involves fewer parameters than the other algorithms discussed above. However, CS-LBG is slower (around 1.425 periods), then the correlation between HBMO-LBG and FA-LBG. Slow merging leads to imperfection by the proposed technique, but it will be enhanced by changes in the algorithm to a certain extent.

(g) KFGC VQ Technique

The authors [29] have proposed an algorithm that practices a sorting manner to generate the codebook, and code vectors have performed the median pattern. This algorithm generates slabs from an image and transforms them to size a vector M $\times$ a range where M is the amount of image formulating vectors of size a. Sorting of fitting vectors has been calculated using initial values of whole vectors.

Let Y= {Y_1, Y_2, ... , Y_t} be the training sequence of M source vectors. The source vector is of dimension Z, Z_i = {$a_{i,1}$, $a_{i,2}$ . . . . . . $a_{i,Z}$} for i = 1,2, . . . , M.

Let P be the set of R clusters for R code vectors, i.e., P(i) denotes i$^{\mathrm{th}}$ code vector for $i = 1,\ 2,\ \ldots.,\ N$. Let Mean Square Error (MSE) of N clusters for each set, i.e., MSE(i) denote the i$^{\mathrm{th}}$ cluster in MSE:

$$\mathrm{MSE}(1) = \frac{1}{\mathrm{MZ}} \sum\nolimits_{t}^{M} \|Y_t - \ D_1\|^2 \tag{32}$$

where, D_1 is the code vector from codebook.

After that, codevectors b_t and b_{t+1} can be computed by taking mean of all vectors in set $P(t)$ and $P(t+1)$ respectively. Compute the mean square error by:

$$\mathrm{MSE}(t) = \frac{1}{\mathrm{VK}} \sum\nolimits_{i=1}^{V} \|Y_s - \ D_m\|^2 \tag{33}$$

where V is the total number of vectors in P(t) and Y_s is the vector in the cluster P(t) for s = 1, 2, . . . , V.

At last, a quick sort algorithm is applied with outcomes to generate a codebook [30]. Table 1 presents different types of VQ techniques for image compression (Imp Comp) with their descriptions, advantages, and disadvantages.

Table 1. Different Types of VQ Techniques for Image Compression.

Ref No.	Techniques	Description	Advantages	Disadvantages
[2]	PSO-LBG	It uses the collective movement attributed to colony of birds or shoals of fish. This has the ability to foster a VQ codebook by augmented PSO flock method, in which the utilized PSO approach processed by LBG technique engenders a VQ codebook used for Img Comp. Regenerated images have improved condition than in outmoded LBG approach.	Additional meticulous computation with active computing time as compared to LBG by updating the h_{best} and q_{best} solution.	Does not cope with the inconsistency in particles.
[2]	HBMO—LBG	It mirrors the behavior of an egg-laying queen bee. It is an improvement over the outdated LBG, PSO-LBG and QPSO-LBG and this approach can furnish a tremendous codebook with slight deformity.	Regenerated image of excellent condition along with an augmented codebook with slight deformity compared to PSO-LBG, QPSO-LBG and LBG approaches.	It does not allow for the slight deformity.
[2]	FA-LBG	To further increase the image quality, optimization techniques, i.e., firefly algorithm (FA)-LBG method are used to optimize and generate the global codebook. In FA-LBG, good quality images are retrieved using the optimized PSO techniques.	FA- LBG needs very minimal computation time and total count of parameters is also minimal as compared to other approaches.	This technique subjected to complication when there are no brighter fireflies in the quest area.
[2]	CS-LBG	Cuckoos birds lay eggs in other birds' nests. If the owner bird identifies those offsprings are not its own, it ejects those nestlings or vacates the nest and seeks a new nest. Additionally, Cuckoo Search seeks for the local and overall codebook using controller limit called mutation probability (Qa). Mutation probability of 0.25 begins native codebook with 25% of merging period along overall codebook receipts 75% of merging period. CS-LBG is 1.425 periods more time-consuming than HBMO-LBG and FA-LBG when merged.	The PSNR rate and quality of the remade image achieved with CS techniques are greater to those acquired with other techniques mentioned above.	CS-LBG is about 1.425 times more sluggish on the uptake in merging as compared to HBMO-LBG and FA-LBG.
[28]	KFCG	This algorithm proved better than LBG algorithm. It used sorting techniques and median approaches to generate codebook.	It is a more effective method as compared to LBG and justified that method gives negligible MSE ratio and tremendous PSNR ratio, which included a short calculation time.	KFGC works better than LBG algorithm on different codebook sizes but the same efficiency cannot be retrieved in real-time videos or pictures.

3. Proposed CS-KFGC Vector Quantization Method

In this section, the proposed CS-KFGC VQ method is explained. After analyzing all the algorithms and their theories, every algorithm has some advanced features that are more proficient than CS-LBG [2] with its drawn-out union speed as compared to other algorithms.

In the proposed model, a combination of Cuckoo Search and Kekre's algorithm [28] is used, excluding the LBG algorithm from CS-LBG. In [28], the author proposed that Kekre's algorithm is more efficient than the LBG algorithms, but the CS-KFGC VQ method yields better results than both Kekre's and LBG algorithms. The KFGC VQ method used a median approach for generating codebooks. Here, Cuckoo Search optimization nests will be initialized and an image with $n * n$ dimensions is selected for input. The minimum adequacy value has been calculated after training the codebook.

The steps for the proposed CS-KFGC VQ method are:

Step 1: Original image of n ∗ n dimensions is taken as input image.

Step 2: Number of nests for Cuckoo Search optimization is initialized.

Step 3: After initializing nests, a minimum adequacy value, i.e., f_{min} is calculated.

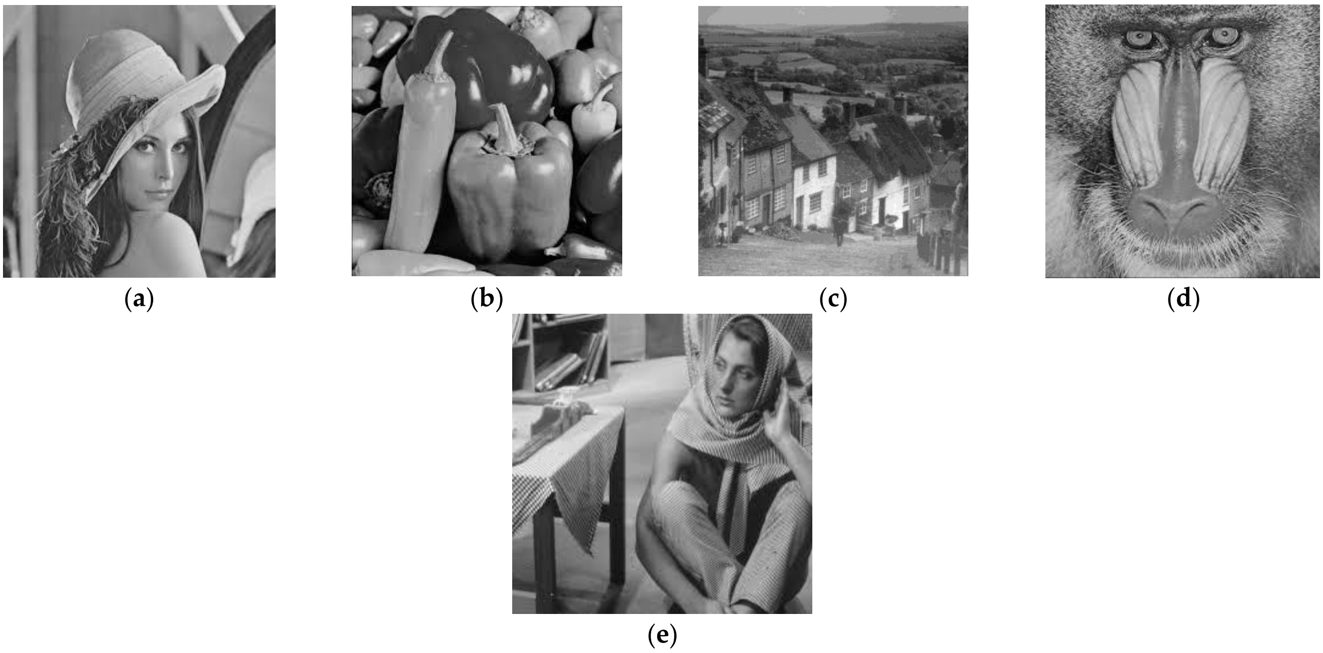
Step 4: Load the KFCG algorithm and train the codebook for VQ.

Step 5: Preprocess the image and test the VQ. Peak Signal-to-Noise (PSNR) value, and computational time is calculated for all methods by compression and decompression of the image.

By implementing the CS-KFGC VQ method, higher PSNR value and lesser computational time are achieved.

4. Results

After applying the proposed CS-KFGC algorithm, different images, i.e., “LENA”, ”PEPPERS”, “GOLDHILL”, “BABOON” and “BARB” were selected for comparison purposes as shown in Figure 2. Gray scale images were used for the codebook design and a size of 512 × 512 pixels was used for the comparison. All the images were in .jpg format except the "PEPPERS" image was in .png format. Images used in CS-LBG, FA-LBG, HBMO-LBG, QPSO-LBG, PSO-LBG and LBG were compressed before applying the algorithm. The implementation was executed using MATLAB tool 2014a with an Intel core i3 processor on a B560 Lenovo laptop. Compressed images were further subdivided into size 4 × 4 pixels with non-coinciding blocks. All the subdivided images were blocks that can be treated as training vectors of (4 × 4) 16 blocks and then a codebook was produced. Decompressed images of “LENA”, ”PEPPERS”, “GOLDHILL”“ BABOON”and “BARB” with respect to 8, 16, 32, 64, 128, 256, 512, 1024 codebook sizes are shown in Figures 3–7.



(**a**) (**b**) (**c**) (**d**) (**e**)

Figure 2. The five test images: (**a**) LENA Image, (**b**) PEPPERS Image, (**c**) GOLDHILL Image, (**d**) BABOON Image and (**e**) BARB Image.

The input vector that is encoded using a codebook is $16,834 = \left(\frac{512}{4} \times \frac{512}{4}\right)$. The result for the proposed CS-KFGC algorithm was produced by calculating PSNR and bits per pixel (bpp) and both were matched with different algorithms. The PSNR values and the computational time was calculated with varying sizes of codebook, i.e., 8, 16, 32, 64, 128, 256, 512, and 1024.The characteristic of the reconstructed image and calculation of data dimension of the compressed image with differing codebook intensity, i.e., 8, 16, 32, 64, 128, 256, 512 and 1024, can be assessed by PSNR:

$$\mathrm{bpp} = \frac{\log_2 \mathrm{N_a}}{\mathrm{m}} \tag{34}$$

where $\mathrm{N_a}$ is the codebook size and m is the intensity of a slab. The comparison with respect to codebook intensities and PSNR values is shown in Figures 8–12.

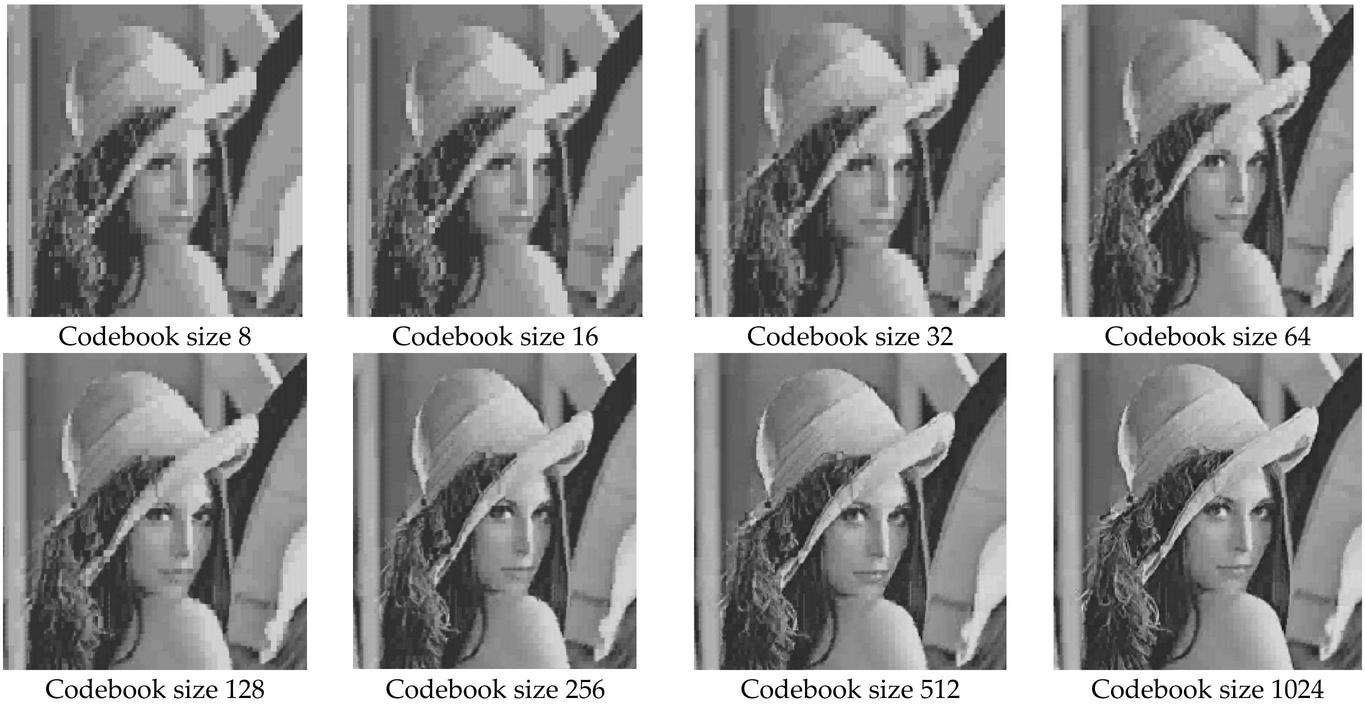

Codebook size 8
Codebook size 16
Codebook size 32
Codebook size 64
Codebook size 128
Codebook size 256
Codebook size 512
Codebook size 1024

Figure 3. Decompressed images of LENA with respect to codebook sizes.

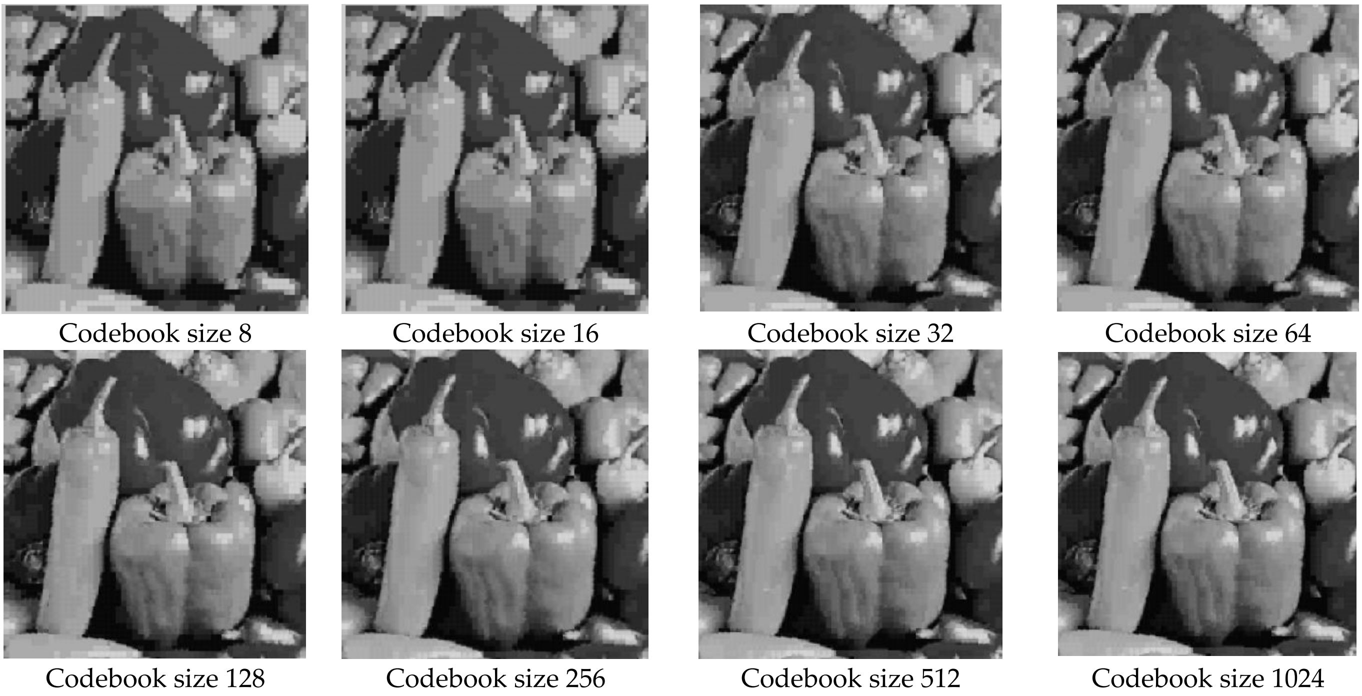
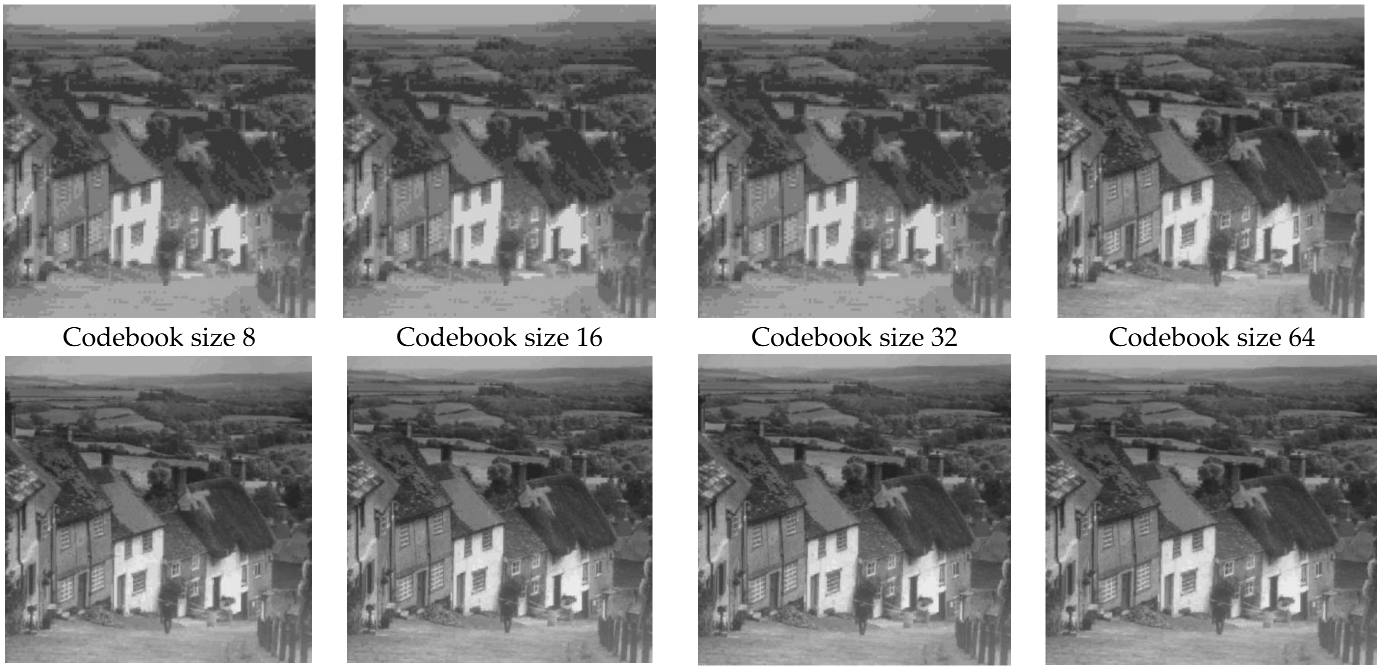

Codebook size 8
Codebook size 16
Codebook size 32
Codebook size 64
Codebook size 128
Codebook size 256
Codebook size 512
Codebook size 1024

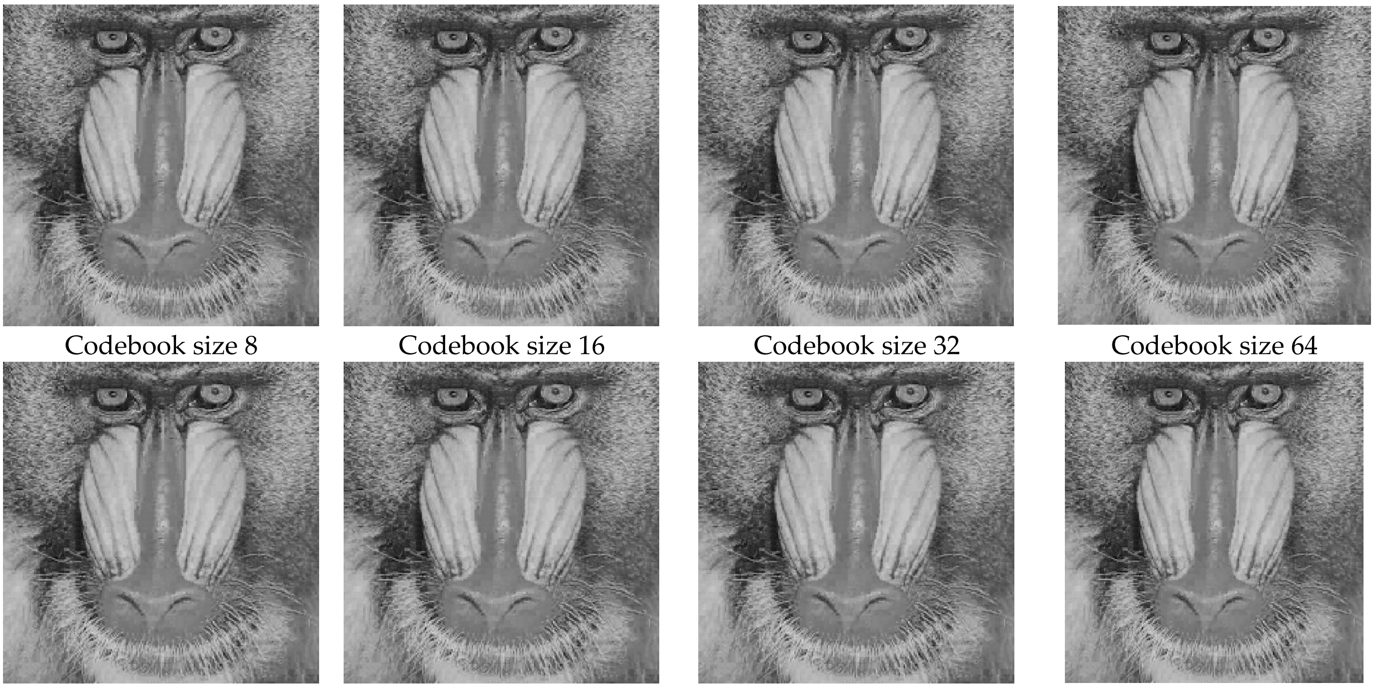
Figure 4. Decompressed images of PEPPERS with respect to codebook size.



Codebook size 8 Codebook size 16 Codebook size 32 Codebook size 64

Codebook size 128 Codebook size 256 Codebook size 512 Codebook size 1024

Figure 5. Decompressed images of GOLDHILL with respect to codebook sizes.



Codebook size 8 Codebook size 16 Codebook size 32 Codebook size 64

Codebook size 128 Codebook size 256 Codebook size 512 Codebook size 1024

Figure 6. Decompressed images of BABOON with respect to codebook sizes.

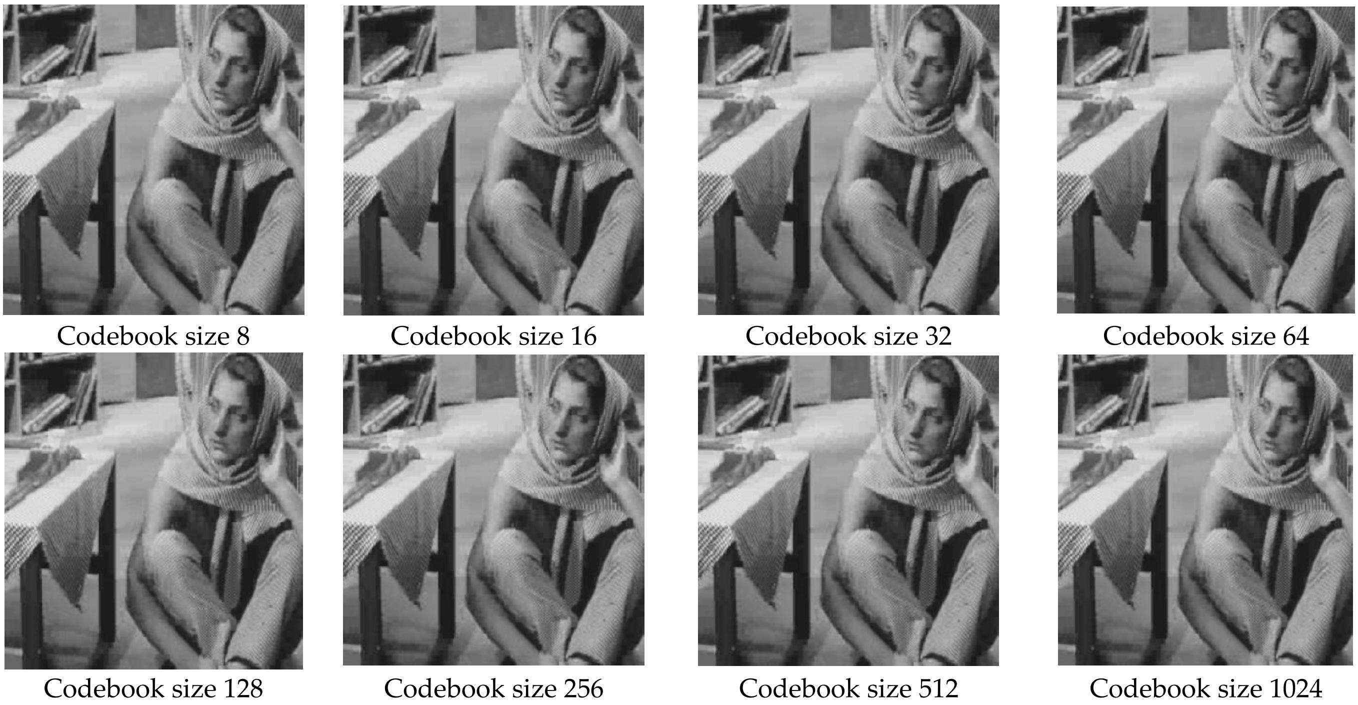

Codebook size 8
Codebook size 16
Codebook size 32
Codebook size 64
Codebook size 128
Codebook size 256
Codebook size 512
Codebook size 1024

Figure 7. Decompressed images of BARB with respect to codebook sizes.

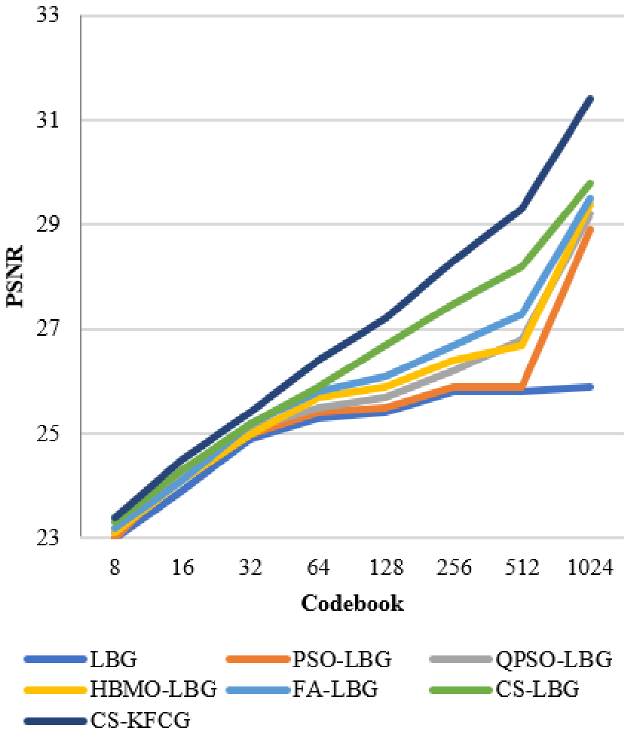


Figure 8. Comparison of PSNR with different VQ methods for “LENA” images.

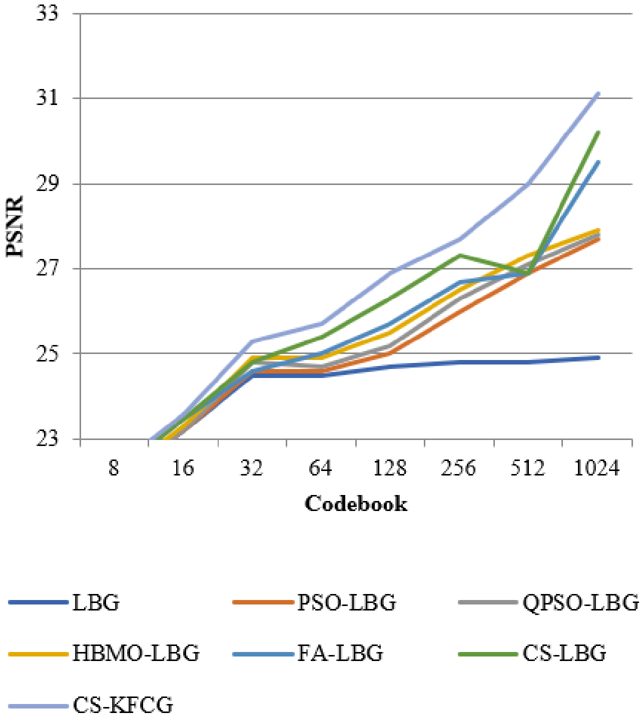


Figure 9. Comparison of PSNR with different VQ methods for "PEPPERS" image.

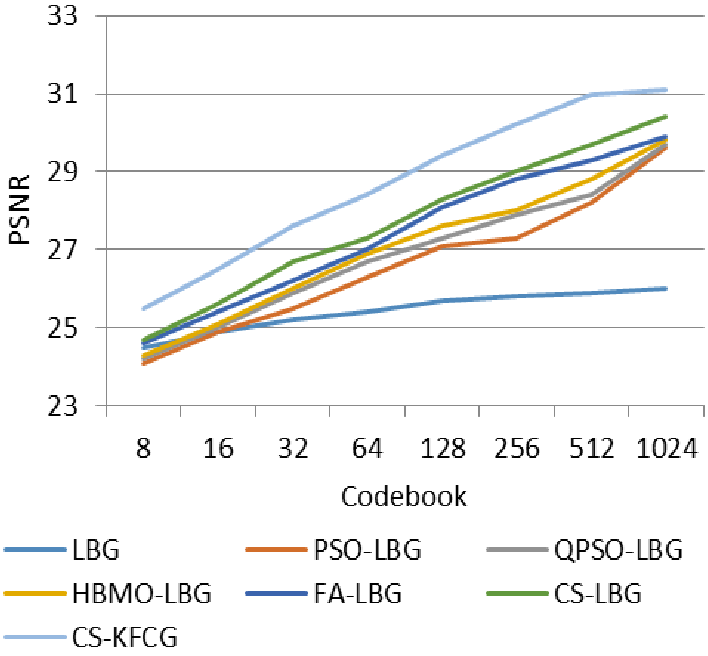


Figure 10. Comparison of PSNR with different VQ methods for "GOLDHILL" images.

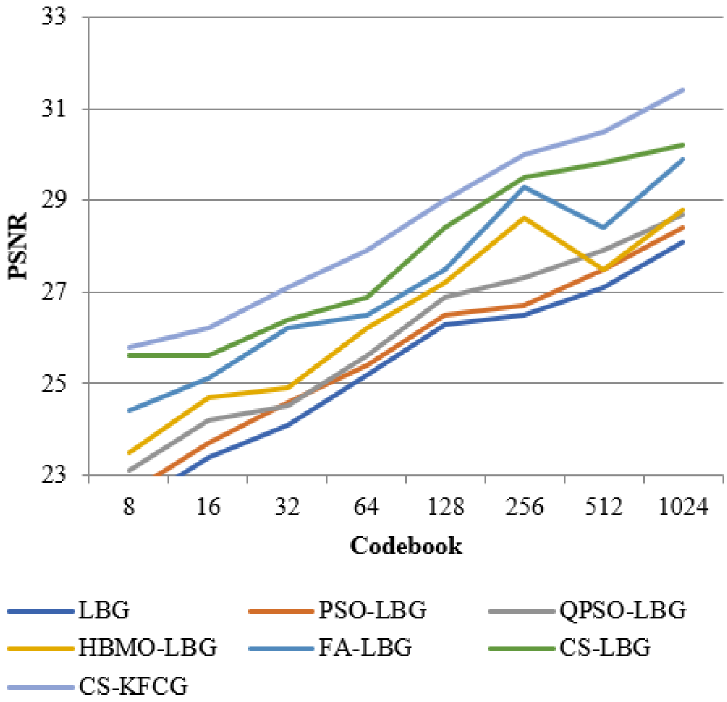


Figure 11. Comparison of PSNR with different VQ methods for "BABOON" images.

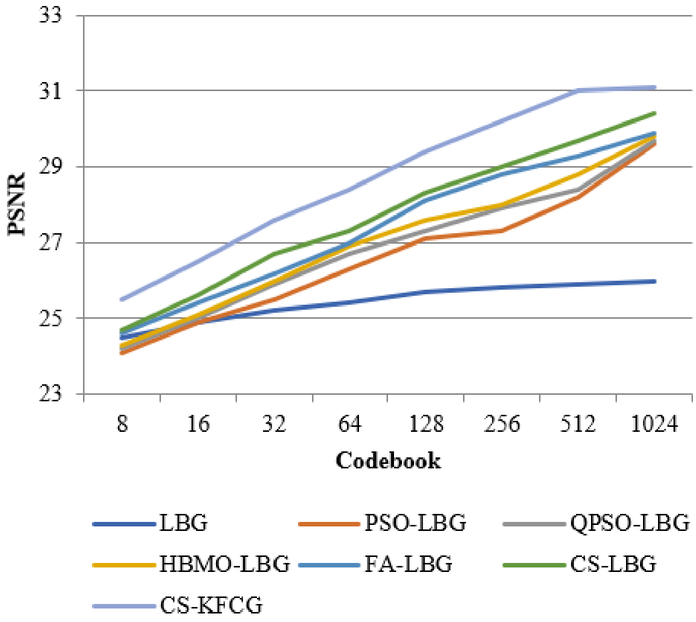


Figure 12. Comparison of PSNR with different VQ methods for "BARB" images.

5. Conclusions

This paper proposed a CS-KFGC algorithm for Img Comp using VQ. By implementing CS-KFGC, the maximum value of PSNR has been achieved. Here, efficient codebook design and vector quantization of training vectors has been investigated using all possible parameters of the CS algorithm. Using efficient training vectors and codebook design for VQ, the proposed method gives excellent results by varying all feasible limits of Cuckoo Search. With mutation expectations and skewness parameters considered, the algorithm's growth and modification are achieved convincingly. Intensification selects the best result from all the results, and diversification yields the inspection of the quest area accurately through randomization. Compared with the CS algorithm, a high peak signal-to-noise ratio and reconstructed image quality have been observed. It is recognized that after practicing the CS-KFGC algorithm, PSNR and computational time of the recreated image yield superior results to those acquired with LBG, PSO-LBG, QPSO-LBG, HBMO-LBG, FA-LBG, etc. The primary limitation of the proposed method is its convergence speed is very slow, but alterations will upgrade it in future.

Author Contributions: Conceptualization: A.B., P.N.B., A.G. and S.T.; writing—original draft preparation: A.T., M.S.R., F.A. and A.G.; methodology: F.A., A.T., G.S. and S.T.; writing—review and editing: S.T., M.S.R., A.B. and A.T.; Software: P.N.B., A.B., G.S. and S.T.; Visualization: A.G., M.S.R., G.S. and P.N.B.; Investigation: A.G., F.A., A.T. and S.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Researchers Supporting Project number (RSP2023R509) King Saud University, Riyadh, Saudi Arabia and also supported by the University of Johannesburg, Johannesburg, South Africa.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

VQ	Vector Quantization
Img Comp	Image Compression
LBG	Linde–Buzo–Gray
PSO-LBG	Particle Swarm Optimization—Linde–Buzo–Gray
QPSO-LBG	Quantum Particle Swarm Optimization–Linde–Buzo–Gray
HBMO-LBG	Honeybee Mating–Linde–Buzo–Gray
FA-LBG	Firefly Algorithm–Linde–Buzo–Gray
CS-LBG	Cuckoo Search–Linde–Buzo–Gray
CS-KFGC	Cuckoo Search–Kekre Fast Codebook Generation
PSNR	Peak Signal-To-Noise Ratio
SQ	Scalar Quantization

References

1. Linde, Y.; Buzo, A.; Gray, R.M. An algorithm for vector quantize design. *IEEE Trans. Commun.* **1980**, *28*, 84–95. [CrossRef]
2. Karri, C.; Jena, U.R. Image compression based on vector quantization using cuckoo search optimization technique. *Ain Shams Eng. J.* **2018**, *9*, 1417–1431.
3. Patane, G.; Russo, M. The enhanced LBG algorithm. *Neural Netw.* **2001**, *14*, 1219–1237. [CrossRef] [PubMed]
4. Wang, Y.; Feng, X.; Huang, Y.; Pu, D.; Zhou, W. A novel quantum swarm evolutionary algorithm and its applications. *Neurocomputing* **2007**, *70*, 633–640. [CrossRef]
5. Karaboga, D.; Basturk, B. A Novel Clustering Approach: Artificial Bee colony (ABC) algorithm. *Appl. Soft Comput.* **2011**, *11*, 652–657. [CrossRef]
6. Horng, M.-H.; Jiang, T.-W. The Artificial Bee Colony Algorithm for Vector Quantization in Image Compression. In Proceedings of the IEEE 4th IEEE International Conference on Broadband Network and Multimedia Technology, Shenzhen, China, 28–30 October 2011; pp. 319–323.
7. Horng, M.H. Honey bee mating optimization vector quantization scheme in image compression. In *Artificial Intelligence and Computational Intelligence*; Deng, H., Wang, L., Wang, F.L., Lei, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5855, pp. 185–194.
8. Yang, X.S. *Nature-Inspired Metaheuristic Algorithms*; Luniver Press: London, UK, 2008.
9. Yang, X.S. Firefly algorithms for multimodal optimization in stochastic algorithms: Foundation and applications. *Lect. Notes Comput. Sci.* **2009**, *5792*, 169–178.
10. Zhou, Y.; Kwong, S.; Guo, H.; Zhang, X.; Zhang, Q. A two-phase evolutionary approach for compressive sensing reconstruction. *IEEE Trans. Cybern.* **2017**, *47*, 2651–2663. [CrossRef] [PubMed]
11. Zhou, Y.; Kwong, S.; Zhang, Q.; Wu, M. Adaptive patch-based sparsity estimation for image via MOEA/D. In Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, 24–29 July 2016.

12. Zhou, Y.; Qiu, Y.; Kwong, S. Region Purity-based Local Feature Selection: A Multi-Objective Perspective. *IEEE Trans. Evol. Comput.* early access. **2022**. [CrossRef]
13. Zhou, Y.; Zhang, W.; Kang, J.; Zhang, X.; Wang, X. A problem-specific non-dominated sorting genetic algorithm for supervised feature selection. *Inf. Sci.* **2021**, *547*, 841–859. [CrossRef]
14. Chang, C.-C.; Chou, J.-S.; Chen, T.-S. An efficient computation of Euclidean distances using approximated look-up table. IEEE Trans. *Circuits Syst. Video Technol.* **2000**, *10*, 594–599. [CrossRef]
15. Zhou, Y.; Kang, J.; Kwong, S.; Wang, X.; Zhang, Q. An evolutionary multi-objective optimization framework of discretization-based feature selection for classification. *Swarm Evol. Comput.* **2021**, *60*, 100770. [CrossRef]
16. Zhou, Y.; Kang, J.; Guo, H. Many-objective optimization of feature selection based on two-level particle cooperation. *Inf. Sci.* **2020**, *532*, 91–109. [CrossRef]
17. Aditya, B.; Gupta, S. An efficient face anti-spoofing and detection model using image quality assessment parameters. *Multimed. Tools Appl.* **2022**, *81*, 35047–35068.
18. Rausheen, B.; Bakshi, A.; Gupta, S. Performance evaluation of optimization techniques with vector quantization used for image compression. In *Harmony Search and Nature Inspired Optimization Algorithms: Theory and Applications*; Springer: Singapore, 2019.
19. Bakshi, A.; Gupta, S.; Gupta, A.; Tanwar, S.; Hsiao, K.-F. 3T-FASDM: Linear discriminant analysis-based three-tier face anti-spoofing detection model using support vector machine. *Int. J. Commun. Syst.* **2020**, *33*, e4441. [CrossRef]
20. Bakshi, A.; Gupta, S. Face Anti-Spoofing System using Motion and Similarity Feature Elimination under Spoof Attacks. *Int. Arab. J. Inf. Technol.* **2022**, *19*, 747–758. [CrossRef]
21. Bakshi, A.; Gupta, S. A taxonomy on biometric security and its applications. Innovations in Information and Communication Technologies. In *Proceedings of International Conference on ICRIHE-2020, Delhi, India: IICT-2020*; Springer International Publishing: Cham, Switzerland, 2021.
22. Chen, Q.; Yang, J.; Gou, J. Image Compression Method Using Improved PSO Vector Quantization. In *Advances in Natural Computation*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 490–495.
23. Yang, X.S.; Deb, S. Cuckoo search via levey flights. In Proceedings of the World Congress on Nature and Biologically Inspired Computing, Coimbatore, India, 9–11 December 2009; Volume 4, pp. 210–214.
24. Yang, X.; Sand, D.S. Engineering optimization by cuckoo search. *Int. J. Math Model Numer. Optim.* **2010**, *4*, 330–343.
25. Chakraverty, S.; Kumar, A. Design optimization for reliable embedded system using cuckoo search. In Proceedings of the International Conference on Electronics Computer Technology, Kanyakumari, India, 8–10 April 2011; Volume 1, pp. 264–268.
26. Valian, E.; Mohanna, S.; Tavakoli, S. Improved cuckoo search algorithm global optimization. *Int. J. Commun. Inf. Technol.* **2011**, *1*, 3–44.
27. Payne, R.B.; Sorenson, M.D.; Klitz, K. *The Cuckoos*; Oxford University Press: Oxford, UK, 2005.
28. Blum, C.; Roli, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *J. Comput. Surv. ACM Digit Lib* **2003**, *35*, 268–308. [CrossRef]
29. Kekre, H.B.; Sarode, T.K. An Efficient Fast Algorithm to Generate Codebook for Vector Quantization. In Proceedings of the ICETET 2008 First IEEE International Conference on Emerging Trends in Engineering and Technology, Maharashtra, India, 16–18 July 2008; pp. 62–67.
30. Kekre, H.B.; Sarode, T.K.; Sange, S.R.; Natu, S.; Natu, P. Halftone image data compression using Kekre's fast code book generation (KFCG) algorithm for vector quantization. In Proceedings of the Technology Systems and Management: First International Conference, ICTSM 2011, Mumbai, India, 25–27 February 2011; Selected Papers. Springer: Berlin/Heidelberg, Germany, 2011.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.