applied sciences

MDPI

Article

# A Texture Classification Approach Based on the Integrated Optimization for Parameters and Features of Gabor Filter via Hybrid Ant Lion Optimizer

**Mingwei Wang [1,*], Lang Gao [2], Xiaohui Huang [2], Ying Jiang [3] and Xianjun Gao [4]**

1 Institute of Geological Survey, China University of Geosciences, Wuhan 430074, China
2 School of Computer Science, China University of Geosciences, Wuhan 430074, China; langgao@cug.edu.cn (L.G.); xhhuang@cug.edu.cn (X.H.)
3 Changjiang River Scientific Research Institute, Changjiang Water Resources Commission, Wuhan 430010, China; jiangying@mail.crsri.cn
4 School of Geoscience, Yangtze University, Wuhan 430100, China; junxgao@whu.edu.cn
* Correspondence: wangmingwei@cug.edu.cn; Tel.: +86-137-2012-1823

Received: 15 March 2019; Accepted: 16 May 2019; Published: 28 May 2019


**Abstract:** Texture classification is an important topic for many applications in machine vision and image analysis, and Gabor filter is considered one of the most efficient tools for analyzing texture features at multiple orientations and scales. However, the parameter settings of each filter are crucial for obtaining accurate results, and they may not be adaptable to different kinds of texture features. Moreover, there is redundant information included in the process of texture feature extraction that contributes little to the classification. In this paper, a new texture classification technique is detailed. The approach is based on the integrated optimization of the parameters and features of Gabor filter, and obtaining satisfactory parameters and the best feature subset is viewed as a combinatorial optimization problem that can be solved by maximizing the objective function using hybrid ant lion optimizer (HALO). Experimental results, particularly fitness values, demonstrate that HALO is more effective than the other algorithms discussed in this paper, and the optimal parameters and features of Gabor filter are balanced between efficiency and accuracy. The method is feasible, reasonable, and can be utilized for practical applications of texture classification.

**Keywords:** texture classification; Gabor filter; parameter optimization; feature selection; hybrid ant lion optimizer

## 1. Introduction

Texture [1,2] is a core property of object appearance in natural scenes, ranging from large-scale samples to microscopic ones. It is also active visual information that is used to describe and recognize objects in the real environment. Texture classification is one of the key problems in texture analysis, and it has been a long-standing research topic because of its importance in understanding the process of texture classification by humans and its extensive applications in computer vision and image analysis [3]. The main applications of texture classification include understanding medical images, extracting visible objects, retrieving content-based images, inspecting industrial faults [4–7], and so on.

In general, the primary focus of studies on texture classification has been the determination of methods to extract texture features. It is generally believed that the extraction of powerful texture features is more important than that of weak texture features since they do not lead to good classification results, even when using excellent classifiers [8]. However, texture features can be found in various orientations and at different scales, and these cannot be characterized effectively by commonly used methods [9–12]. Gabor filter has been used for this purpose and performs better in

the discrimination of individual texture features, especially those with similar descriptions. However, a single filter is difficult to apply to multi-orientation and multiscale texture features. Thus, a bank of Gabor filters with the ability to extract multi-orientation and multiscale texture features was proposed to address the issue. Gabor filter has been extensively used for texture feature extraction and texture classification [13]. Li [14] captured the dependence of Gabor filter on different channels, and had better performance results than that of state-of-the-art approaches. Younesi [15] proposed a palm print recognition method that used a bank of Gabor filters to extract texture features from images, and the method achieved higher accuracy than texture classification approaches based on single Gabor filter. Huang [16] proposed a new technique of identifying group-housed pigs on the basis of Gabor filter, and the experimental results demonstrate that the accuracy outperformed the compared approach by 91.86%, and PCA parameter drifted in the range of 0.85–0.9. Lu [17] presented a texture classification method for fracture risk estimation based on Gabor filter combined with the total T-score, and the shape texture features could be measured in a wider area compared with other methods. Kim [18] proposed a novel texture classification technique with directional statistical (DS)-Gabor filter and had satisfactory rotation invariance owing to the combination of a number of directional statistics. However, the parameters of the above method need to be set by experience, thus it may not be adaptable to different types of texture features. Furthermore, the parameter settings of each Gabor filter are crucial to classification accuracy.

On the other hand, the process of obtaining the optimal parameters of Gabor filter can be seen as a combinatorial optimization problem that can be tackled by swarm intelligence algorithms. For example, Khan [19,20] presented a technique for the optimization of each filter individually via particle swarm optimization (PSO) and cuckoo search (CS) algorithm, and the approach successfully represents local texture changes at multiple scales and orientations from mammograms with an effective improvement in classification accuracy. Tong [21] proposed a defect detection technique via Gabor filter to inspect flaws in woven fabrics in the fashion industry, and the differential evolution (DE) algorithm was utilized to obtain the optimal parameters of the filter bank. The method achieved high successful detection and low false alarm rates. However, the optimization ability of PSO, DE, and CS algorithms is mainly based on a random search, which cannot ensure convergence toward the optimal solution. Moreover, because of the multi-orientation and multiscale nature of the filter bank, there is redundant information in the process of texture feature extraction, and it contributes little to the classification [22]. Thus, numerous feature selection approaches based on Gabor filter have been proposed to reduce the data dimension [23,24], but the previous works have ignored the relevance of parameter optimization and feature selection. In essence, parameter optimization and feature selection are both considered combinatorial optimization problems, and the integrated optimization of the parameters and features of Gabor filter could be obtained by swarm intelligence algorithms at the same time.

Ant lion optimizer (ALO) [25] is a novel swarm intelligence algorithm. Nowadays, ALO has been applied in various fields, such as power system design, fault detection, schedule planning, path searching and so on [26–29]. The optimization ability of ALO does not rely on any parameters, and it is more likely to obtain the satisfactory solution. However, feature selection is considered a discrete optimization problem and is difficult to solve using ALO with decimal coding. Mafarja [30] presented a binary coded ALO (BALO) to select the optimal feature subset for some well-known datasets from the UCI repository. BALO obtained superior results by searching for the best feature subset, and the performance was independent of the step-length and classifier. Hence, in this paper, a new texture classification method is proposed that blends the use of Gabor filter and a hybrid binary-decimal coded ALO (HALO) to, respectively, solve the problems of parameter optimization and feature selection.

The rest of this paper is structured as follow. The basic principle of HALO is illustrated in Section 2. In Section 3, the proposed method to obtain the integrated optimization of the parameters and features of Gabor filter is detailed. Section 4 displays the experimental results and discussion. Finally, the paper is concluded in Section 5.

## 2. Overview of HALO

### 2.1. ALO for Parameter Optimization

In 2015, Mirjalili introduced a swarm intelligence algorithm called ALO, which imitates the hunting behavior of antlions and has no parameters to be set [25]. In nature, an antlion larva digs a cone-shaped pit in the sand by moving along a circular path and throwing out sand with its massive jaw, and the larva then hides underneath the bottom of the cone and waits for ants to be trapped in the pit. The exploratory behavior of ALO is similar to an antlion's digging in a circular path. Additionally, the exploitation behavior of ALO is similar to the boundary adjustment of antlions' traps. As each ant in ALO moves randomly in the solution space, its strategy is similar to that of spinning a roulette wheel and is represented below:

$$X^t = [0, cumsum(2r(t_1) - 1), cumsum(2r(t_2) - 1), ..., cumsum(2r(t_n) - 1)] \quad (1)$$

where $X^t$ is the position of ants after a random walk, $n$ is the number of ants in the population, $cumsum$ represents the cumulative sum, $t$ is the iteration number, and $r(t)$ is a stochastic distribution function as defined below:

$$r(t) = \begin{cases} 1 & if\ rand > 0.5 \\ 0 & if\ rand \leq 0.5 \end{cases} \quad (2)$$

where $rand$ is a random number taken from a uniform distribution in the interval of [0,1].

To maintain the random walk in the scope of the search space, the positions of each ant are normalized by min-max normalization as defined by the following:

$$X_i^t = \frac{(X_i^t - a_i)(b_i - c_i^t)}{d_t^t - a_i} + c_i^t \quad (3)$$

where $a_i$ is the minimum of the random walk of $i$th variable, $b_i$ is the maximum of the random walk of $i$th variable, $c_i^t$ is the minimum of $i$th variable at iteration $t$, and $d_i^t$ indicates the maximum of $i$th variable at iteration $t$.

Antlions build traps that are proportional to the size of the solution space, and the ants move stochastically. Moreover, antlions shoot sand outward from the center of the pit once they realize that an ant is in the trap to keep the ant trapped as it tries to escape. To mathematically define this behavior, the radius of an ant's random walk on a hypersphere is reduced adaptively, and the process of trapping can be described as follows:

$$c^t = \frac{c^t}{I} \quad (4)$$

$$d^t = \frac{d^t}{I} \quad (5)$$

where $I = 10^{\omega} \frac{t}{T}$, $T$ is the maximum number of iterations, and $\omega$ is a constant variable that is defined based on the current iteration ( $\omega = 2$ when $t > 0.1\ T$, $\omega = 3$ when $t > 0.5\ T$, $\omega = 4$ when $t > 0.75\ T$, $\omega = 5$ when $t > 0.9\ T$, and $\omega = 6$ when $t > 0.95\ T$). Basically, $\omega$ is utilized to adjust the optimization level of exploration.

### 2.2. Binary Coding for Feature Selection

In general, feature selection is regarded as a discrete combinatorial optimization problem and is difficult to solve directly by decimal coding. In the binary coding environment, moving through each

band means that the position of each individual changes from 0 to 1 or vice versa [31]. To introduce a binary coding form of ALO, the coding process of each ant can be seen as continuous optimization of similarity. The major difference between the decimal and binary coding form is that ants are characterized by a realistic value or by switching to "0" or "1" in the transform function. That is to say, the position of each ant calculated by Equation (3) is set to either 0 or 1 with a certain probability and calculated with a conditional function, as shown in Equation (6):

$$R_i^t = \begin{cases} 1 & if \quad rand \leq |\tanh(X_i^t)| \\ 0 & otherwise \end{cases} \tag{6}$$

where $tanh(\cdot)$ is the hyperbolic tangent function and $R_i^t$ is the form of binary coding for the position of each ant that is utilized to solve the problem of feature selection.

To summarize, in this paper, HALO is proposed with the combination of binary and decimal coding to, respectively, solve the problems of parameter optimization and feature selection using an ant, and the aim of this approach is to obtain the optimal classification accuracy for different texture features. The overall operating process of the proposed texture classification method is expressed in the following section.

## 3. The Proposed Method

An efficient texture classification method that applies the integrated optimization of the parameters and features of Gabor filter using HALO is presented in this section. The main goal of texture classification is to precisely classify several texture features in each sample.

### 3.1. Fundamental of Gabor Filter

Gabor filter is calculated by a convolutional kernel function, which has been utilized in image processing applications, especially for texture classification. To represent multiscale and multi-orientation texture features, a filter bank is structured with a number of parameters in different orientations and scales. The essential variable $g(x, y)$ of a 2-D Gabor filter is expressed by a Gaussian kernel and adjusted by a complex sinusoidal wave: [32]:

$$g(x,y) = \left(\frac{1}{2\pi\sigma_x\sigma_y}\right) exp\left(-\frac{1}{2}\left(\frac{\bar{x}^2+\bar{y}^2}{\sigma_x^2+\sigma_y^2}\right) + 2\pi jW\bar{x}\right) \tag{7}$$

$$\bar{x} = xcos\theta + ysin\theta \quad \bar{y} = -xsin\theta + ycos\theta \tag{8}$$

where $\sigma_x$ and $\sigma_y$ are the parameters that describe the spread of the current pixel in the neighborhood in which weighted summation occurs, $W$ is the central frequency of the complex sinusoid, and $\theta \in [0, \pi)$ is the orientation of the horizontal to vertical stripes in the equation above.

A bank of Gabor filters includes multiple individuals and can be adjusted for different orientations and frequencies. Gabor filter considered here is GS5O8 (including 40 filters: 5 scales $S \times 8$ orientations $O$ with an initial max frequency equal to 0.2 and an initial orientation set to 0. The parameter settings of the orientations and frequencies for a bank are tuned by the following formulas [33]:

$$Orientation(i) = \frac{(i-1)\pi}{O}, where\ i = 1, 2, ..., O \tag{9}$$

$$Frequency(i) = \frac{f_{max}}{\sqrt{(2)^{i-1}}}, where\ i = 1, 2, ..., S \tag{10}$$

where $O$ and $S$ are, respectively, the orientations and scales in total, and $f_{max}$ is the maximum value of the frequency and is set to $f_{max} = 0.2$.

### 3.2. Encoding Schema of Parameter Optimization and Feature Selection

The key objective of the proposed texture classification method is to achieve satisfactory parameters and features of Gabor filter and create a reasonable mapping between the solution and HALO. The parameter optimization process needs $O + S$ bits to apply coding to the parameters, which are the orientations $\theta$ and frequencies $W$ in Equations (9) and (10). The former $O$ bits represent each orientation, the latter $S$ is each scale, and the ranges of the parameters are, respectively, set to $(0, 1)$ and $(0, 0.2]$ with decimal coding. For feature selection, the length of coding is equal to the total number of features in the database, and here it is set to $O \times S$. For binary coding, every bit is expressed by the fixed value "0" or "1" by Equation (6), where "1" illustrates that the current feature will be chosen for classification, and "0" illustrates that the current feature will be abandoned. Thus, the total coding length of an ant is equal to $O \times S + O + S$. The encoding form of an ant for Gabor filter with $O$ orientations and $S$ scales is shown in Figure 1. The entire code simultaneously indicates the solution of the parameter settings and the feature subset of Gabor filter.


**Figure 1.** The encoding form of an ant for Gabor filter.

### 3.3. Definition of Objective Function

To evaluate the optimization ability of HALO and other evolutionary algorithms, it is necessary to choose a suitable objective function. Because the process of training can be considered prior knowledge, each texture feature is a category. The Fisher criterion has good performance in the calculation of heuristic knowledge. The Fisher criterion aims to maximize the inter-class difference and minimize the intra-class difference, and it is usually applied to assess binary classification problems [34]. For multi-classification problems, the objective function should be improved to synthetically weight the differences between each category. Therefore, it is defined as shown in Equation (11):

$$fit = \frac{\sum_{i=1}^{l-1} \sum_{j=i+1}^{l} |\mu_i - \mu_j|}{\sum_{i=1}^{l} \sigma_i^2} \tag{11}$$

where $\mu_i$ and $\sigma_i$ are, respectively, the average and the standard deviation of the eigenvalues for $i$th category of texture features, and $l$ is the number of classes. A larger value resulting from the above equation demonstrates better-quality parameters and features for Gabor filter.

### 3.4. Implementation of the Proposed Technique

The overall scheme of the proposed texture classification technique is shown below. First, it performs encoding for each ant of HALO, and the process of parameter optimization and feature selection is then constructed by binary and decimal encoding. The result constitutes a feature vector that is input into a classifier and yields a final label for each sample.

The value of the feature vector is composed of different orientations and frequencies of Gabor filter, and each feature vector is normalized to a uniform magnitude in the range of [0,1]. In general, the process of texture classification has two stages: training and testing. The former stage provides a certain extent of prior knowledge for classification, and the latter stage verifies the availability of the current model [35].

Training samples with different scales and rotations are utilized to avoid overfitting for fair performance evaluation. However, at the same time, the volume of the training dataset is increased by the 40 filters. As a result, interval sampling [36] with a length of 8 pixels and max-pooling with a $4 \times 4$ window is utilized for each sample to decrease the data volume in the process of convolution.

- Step 1: Extract the training and testing samples from each database.
- Step 2: For each ant, generate the decimal coding for parameter optimization, and the binary coding for feature selection.
- Step 3: Perform convolution between each training sample and the filter bank.
- Step 4: Reduce the data dimension via interval sampling and max-polling, and compute the fitness value by using Equation (11).
- Step 5: Operation of HALO:

  Step 5-1: Trap ants in antlion's pits by using Equations (4) and (5).
  Step 5-2: Random walk ants by using Equation (3).
  Step 5-3: Change the encoding of feature selection to the binary form by Equation (6);

- Step 6: Perform convolution between each training sample and the filter bank and build the feature subset according to the binary encoding.
- Step 7: Reduce the data dimension via interval sampling and max-polling, and compute the fitness value by using Equation (11).
- Step 8: If the solution is better, replace the position of the current ant; otherwise, do not change the position and find the current global best solution.
- Step 9: Judge whether the terminal criterion is satisfied: if it is, go to Step 10; otherwise, go to Step 5.
- Step 10: Output the optimal feature subset, and conduct classification for testing samples.

## 4. Experimental Results and Discussion

The proposed method was implemented in the language of MATLAB 2014b on a personal computer with a 2.30 GHz CPU and 8.00 GB RAM on a Windows 8 operating system. To assess the quality of the proposed technique, public databases were utilized to extract features based on Gabor filter with several orientations and frequencies, as described in this section.

### *4.1. Databases Description*

To evaluate the performance of the proposed texture classification method optimized by HALO, three public texture databases were used in the experiment. The first database CGT [37] offers digital pictures of all sorts of materials with the pictures of fabric, wood, metal, bricks, plastic, and these texture images can be used for graphic design and visual effects. In the experiments, 18 homogeneous texture images from the database were as shown in Figure 2, all chosen texture images without any rotation and 10 images for each class were utilized as training samples, while the other 50 images were used as testing samples.

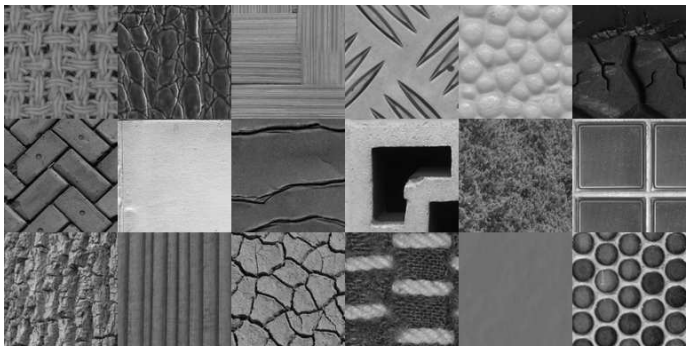

**Figure 2.** Samples of the 18 categories randomly selected from the CGT database.

The second database Kylberg [38] was imaged under only one light setting from one direction on the same distance. Textured surfaces are arranged, such as oatmeals, linseeds, lentils, the texture

samples with the same category have 12 different angles of rotation with 30 degrees increment. In the experiments, 20 homogeneous texture images from the database were as shown in Figure 3, and images without any rotation were utilized as training samples with the number per category set as 15, and 60 images with other angles of rotation per category were used as testing samples.

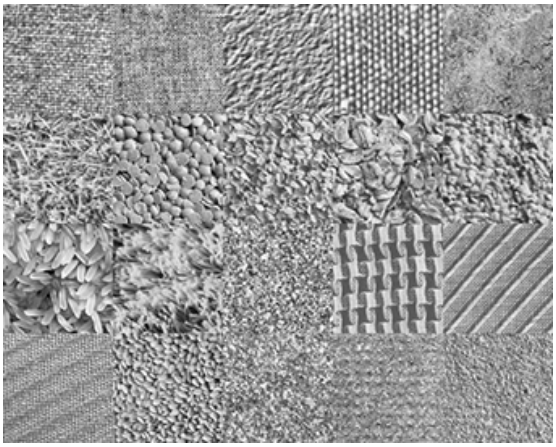

**Figure 3.** Samples of the 20 categories randomly selected from the Kylberg database.

The third database is Brodatz [39] with different background intensities, and the figure below gives an example of 40 different texture features organized into 5 columns. For example, D6 has a black background, whereas D10 has gray and white backgrounds, and D101 is a regular texture, whereas the texture type of D111 is irregular. In the experiments, 30 homogeneous texture images from the database were as shown in Figure 4, all of the images were rotated with a 20° step, images with 20° rotation were utilized as training samples with the number set as 6 for each class. The rest of the 48 images were utilized as testing samples.

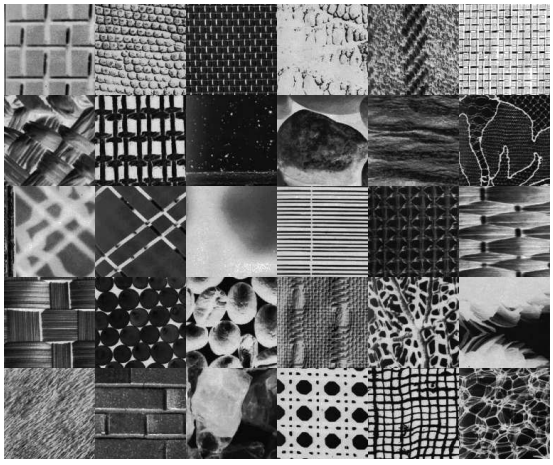

**Figure 4.** Samples of the 30 categories randomly selected from the Brodatz database.

### *4.2. Parameters Setting for Different Algorithms*

As detailed in Section 2, the optimization ability does not rely on any parameter settings in HALO, thus it is prevented from becoming trapped in the local optimal solution to a great extent. In addition, some commonly used swarm intelligence algorithms, such as PSO algorithm [40], DE algorithm [41], CS algorithm [42] and gray wolf optimizer (GWO) [43], were used for an intuition comparison between the optimization abilities. All of these algorithms are based on hybrid decimal and binary coding. Moreover, other types of Gabor filter were used for further comparison. For a relatively fair comparison, the number of function evaluations was used as the terminal criterion; that is, all algorithms stopped when the iteration number reached 20 combined with part of experimental results as the fitness value

did not improve, and all algorithms performed 30 independent operations. Our primary interest was the integrated optimization of parameters and features of Gabor filter, and this was shown by the fitness value of the objective function and the classification accuracy for the testing samples in each database. Table 1 shows the parameter settings of the above algorithms.

**Table 1.** Parameter settings for different algorithms.

| Parameters | Value |
|---|---|
| Population size | 8 |
| Dimension | 40 + 13 |
| Number of runs for each technique | 30 |
| $c_1$, $c_2$ Acceleration coefficient in PSO algorithm | 2.0 |
| $f_m$ Mutation factor in DE algorithm | 0.6 |
| $C_R$ Crossover rate in DE algorithm | 0.9 |
| $G_0$ Initial of gravitational variable in GSA | 100 |
| $\alpha$ User specified constant in GSA | 10 |
| $p_a$ Detecting probability in CS algorithm | 0.25 |
| $\beta$ Parameter in CS algorithm | 1.5 |
| a Correlation coefficient in GWO | [2,0] |
| $r_1$, $r_2$ Random vectors in GWO | [0,1] |

### *4.3. Experiments for Different Swarm Intelligence Algorithms*

A preliminary test of the proposed texture classification approach on three public texture databases, namely CGT, Kylberg, and Brodatz, was conducted, as described here. The size of the training samples was extracted as 128 × 128 with different angles of rotation. In Table 2, *Fiv* is the average fitness value calculated by Equation (11) using the integrated optimization for each filter bank handled by different swarm intelligence algorithms, and *Fn* and Time, respectively, indicate the selected number of features and CPU time, on average, for each training process.

**Table 2.** Result of different algorithms for public texture databases.

| Database | Meas. | PSO | DE | CS | GWO | HALO |
|---|---|---|---|---|---|---|
| CGT | *Fiv* | 15.3503 | 16.1756 | 16.9323 | 17.5238 | 18.3323 |
| | *Fn* | 20.7667 | 20.3333 | 19.8667 | 19.5000 | 19.0333 |
| | Time | 11.2552 | 10.9532 | 10.8711 | 11.0332 | 10.6732 |
| Kylberg | *Fiv* | 15.9803 | 16.8198 | 18.1516 | 19.2445 | 19.7469 |
| | *Fn* | 20.8667 | 20.4333 | 20.2000 | 19.6667 | 19.2000 |
| | Time | 15.3124 | 14.9997 | 14.9056 | 15.1336 | 14.6008 |
| Brodatz | *Fiv* | 13.2545 | 14.2234 | 14.9204 | 16.0043 | 16.9607 |
| | *Fn* | 18.7333 | 18.4000 | 17.8333 | 17.2667 | 16.9000 |
| | Time | 25.2872 | 24.9488 | 24.8150 | 25.0316 | 24.5248 |

As shown in Table 2, the average fitness value using the proposed method was the highest for all databases, proving that the optimization ability of HALO was superior to that of PSO, DE, CS, and GWO. The discrimination ability of each category was enhanced since the fitness value exceeded 16. More importantly, the process of ALO is based on the hunting behavior of antlions and has no parameters to be set. Hybrid decimal and binary encoding qA utilized to conduct integrated optimization of the parameters and features of Gabor filter; this strategy improved the exploitation ability compared with the use of only one ant encoding. Moreover, the selected number of features was the lowest among all of the algorithms involved: the number was higher than 20 for the CGT and Kylberg databases using PSO and DE algorithms, and HALO abandoned more than 50% of the redundant features from Gabor filter. From the aspect of operating efficiency, HALO had a faster convergence rate because it had fewer multiplications compared with the other algorithms. The difference in CPU time reached 0.7 s for the Kylberg and Brodatz databases, and HALO only needed

10.6732 s to select the best feature subset by obvious distinction for the CGT database. Overall, it can be deduced that the optimization ability and operating efficiency are improved by using HALO, which has the desired adaptability to obtain suitable parameters and features of Gabor filter.

### *4.4. Application for Texture Classification*

Next, each texture sample was classified by using the optimal parameters and features of Gabor filter. Figures 5–7 indicate the difference in the average eigenvalues between the training and testing samples for the categories with the highest classification accuracy using the selected parameters and features of Gabor filter. Tables 3–5 show the classification accuracy using the newly proposed Log-Gabor filter [44], DS-Gabor filter [18], the only parameter optimization-based Gabor (OP-Gabor) filter, and the proposed method. In the tables, OA and Kappa are the overall classification accuracy and Kappa coefficient, respectively, obtained by using different texture classification methods. The Kappa coefficient is defined below:

$$k = \frac{P_o - P_e}{1 - P_e} \tag{12}$$

where $P_o$ is the relative observed agreement, and $P_e$ is the hypothetical probability of chance agreement.

**Table 3.** Overall classification accuracy and Kappa coefficient for CGT database.

| Category | Log-Gabor | DS-Gabor | OP-Gabor | Proposed | Category | Log-Gabor | DS-Gabor | OP-Gabor | Proposed |
|---|---|---|---|---|---|---|---|---|---|
| Fabric | 96.00 | 96.00 | 96.00 | 100.00 | Brick | 48.00 | 54.00 | 86.00 | 86.00 |
| Leather | 92.00 | 90.00 | 94.00 | 92.00 | Vegetation | 86.00 | 80.00 | 88.00 | 90.00 |
| Wicker | 64.00 | 90.00 | 96.00 | 96.00 | Window | 94.00 | 94.00 | 94.00 | 96.00 |
| Metal | 94.00 | 94.00 | 96.00 | 96.00 | Tree | 70.00 | 60.00 | 70.00 | 90.00 |
| Plastic | 98.00 | 98.00 | 98.00 | 100.00 | Wood | 96.00 | 94.00 | 98.00 | 96.00 |
| Wheel | 46.00 | 66.00 | 78.00 | 86.00 | Soil | 100.00 | 100.00 | 100.00 | 100.00 |
| Road | 88.00 | 74.00 | 92.00 | 90.00 | Wool | 96.00 | 96.00 | 96.00 | 100.00 |
| Paper | 100.00 | 100.00 | 100.00 | 100.00 | Water | 98.00 | 98.00 | 100.00 | 100.00 |
| Decoration | 60.00 | 98.00 | 98.00 | 100.00 | Tile | 94.00 | 76.00 | 96.00 | 98.00 |
| | | | | | OA(%) | 84.44 | 86.56 | 93.11 | 95.33 |
| | | | | | Kappa | 0.8353 | 0.8576 | 0.9271 | 0.9506 |

**Table 4.** Overall classification accuracy and Kappa coefficient for Kylberg database.

| Category | Log-Gabor | DS-Gabor | OP-Gabor | Proposed | Category | Log-Gabor | DS-Gabor | OP-Gabor | Proposed |
|---|---|---|---|---|---|---|---|---|---|
| Blanket | 96.67 | 83.00 | 96.67 | 96.67 | Rice | 75.00 | 81.67 | 86.67 | 90.00 |
| Canvas | 91.67 | 91.67 | 91.67 | 93.33 | Rug | 66.67 | 75.00 | 95.00 | 96.67 |
| Ceiling | 93.33 | 88.33 | 95.00 | 96.67 | Sand | 55.00 | 71.67 | 86.67 | 93.33 |
| Cushion | 86.67 | 81.67 | 88.33 | 91.67 | Scarf | 96.67 | 96.67 | 98.33 | 98.33 |
| Floor | 83.33 | 86.67 | 90.00 | 91.67 | Screen | 93.33 | 93.33 | 95.00 | 100.00 |
| Grass | 70.00 | 58.33 | 76.67 | 90.00 | Seat | 96.67 | 96.67 | 96.67 | 98.33 |
| Lentils | 73.33 | 76.67 | 88.33 | 95.00 | Sesameseed | 81.67 | 93.33 | 100.00 | 100.00 |
| Linseed | 73.33 | 90.00 | 95.00 | 96.67 | Stone | 78.33 | 86.67 | 90.00 | 91.67 |
| Oatmeal | 65.00 | 80.00 | 91.67 | 95.00 | Stoneslab | 35.00 | 38.33 | 56.67 | 73.33 |
| Perlsugar | 83.33 | 90.00 | 91.67 | 91.67 | Wall | 90.00 | 91.67 | 95.00 | 96.67 |
| | | | | | OA(%) | 79.25 | 82.58 | 90.25 | 93.83 |
| | | | | | Kappa | 0.7816 | 0.8167 | 0.8974 | 0.9351 |

**Table 5.** Overall classification accuracy and Kappa coefficient for Brodatz database.

| Category | Log-Gabor | DS-Gabor | OP-Gabor | Proposed | Category | Log-Gabor | DS-Gabor | OP-Gabor | Proposed |
|---|---|---|---|---|---|---|---|---|---|
| D1 | 87.50 | 97.92 | 97.92 | 97.92 | D49 | 100.00 | 100.00 | 100.00 | 100.00 |
| D3 | 75.00 | 97.92 | 97.92 | 97.92 | D52 | 85.42 | 93.75 | 97.92 | 97.92 |
| D6 | 97.92 | 95.83 | 97.92 | 97.92 | D56 | 95.83 | 89.58 | 95.83 | 95.83 |
| D10 | 100.00 | 100.00 | 100.00 | 100.00 | D64 | 85.42 | 87.50 | 97.92 | 97.92 |
| D11 | 97.92 | 97.92 | 100.00 | 100.00 | D66 | 89.58 | 97.92 | 97.92 | 97.92 |
| D14 | 97.92 | 93.75 | 97.92 | 97.92 | D74 | 95.83 | 95.83 | 95.83 | 95.83 |
| D18 | 81.25 | 91.67 | 93.75 | 93.75 | D83 | 95.83 | 97.92 | 97.92 | 97.92 |
| D20 | 91.67 | 95.83 | 95.83 | 95.83 | D87 | 89.58 | 87.50 | 95.83 | 95.83 |
| D25 | 95.83 | 95.83 | 95.83 | 95.83 | D88 | 83.33 | 89.58 | 95.83 | 95.83 |
| D31 | 70.83 | 85.42 | 91.67 | 91.67 | D93 | 87.50 | 91.67 | 93.75 | 93.75 |
| D37 | 85.42 | 95.83 | 95.83 | 95.83 | D94 | 62.50 | 81.25 | 89.58 | 89.58 |
| D41 | 97.92 | 97.92 | 97.92 | 97.92 | D99 | 83.33 | 87.50 | 91.67 | 91.67 |
| D46 | 93.75 | 95.83 | 95.83 | 95.83 | D101 | 91.67 | 97.92 | 100.00 | 100.00 |
| D47 | 93.75 | 95.83 | 95.83 | 95.83 | D104 | 93.75 | 89.58 | 95.83 | 95.83 |
| D48 | 85.42 | 95.83 | 95.83 | 95.83 | D111 | 85.42 | 87.50 | 93.75 | 93.75 |
| | | | | | OA(%) | 89.24 | 93.61 | 96.32 | 96.32 |
| | | | | | Kappa | 0.8886 | 0.9339 | 0.9619 | 0.9619 |

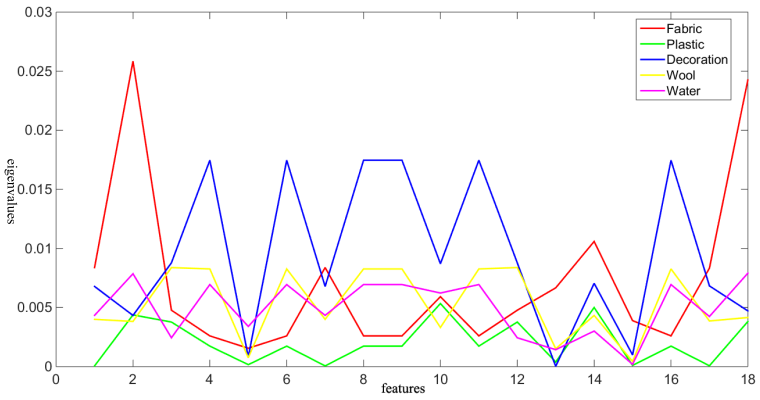


**Figure 5.** Difference of average eigenvalues between training and testing samples for CGT database.

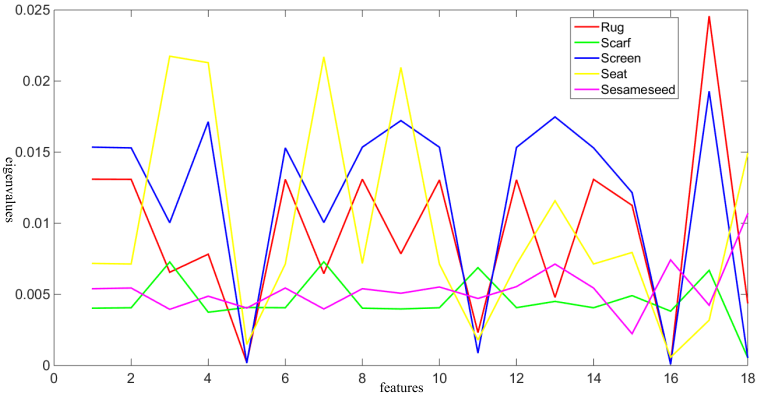


**Figure 6.** Difference of average eigenvalues between training and testing samples for Kylberg database.

Tables 3–5 reveal that the classification accuracy was increased, with a difference of more than 2%, by removing some redundant features of Gabor filter. In addition, more than 40 samples were misclassified for the Kylberg database. Although the classification accuracy for the Leather, Road, and Wood classes in the CGT database was relatively high, misclassification still had a certain influence on the overall process. Moreover, the Kappa coefficient was more than 0.93 for all databases, illustrating that the precision could adapt to a number of application demands. The difference in the average eigenvalues between the training and testing samples was lower than 0.03, and the change trend was similar to that shown in Figures 5–7, thus proving the discriminability and identity of each Gabor filter. The overall classification accuracy of the Log-Gabor filter and DS-Gabor filter was unsatisfactory: it was lower than 85% for the CGT database and only 86.56% and 82.58%, respectively, for the Kylberg

database. With these two methods, very few samples were correctly classified for the Wheel, Brick, Sand, and Stone slab categories, thus it was difficult for them to extract texture features from the whole database. In brief, the proposed approach is a reliable, efficient, and reasonable method for texture classification.

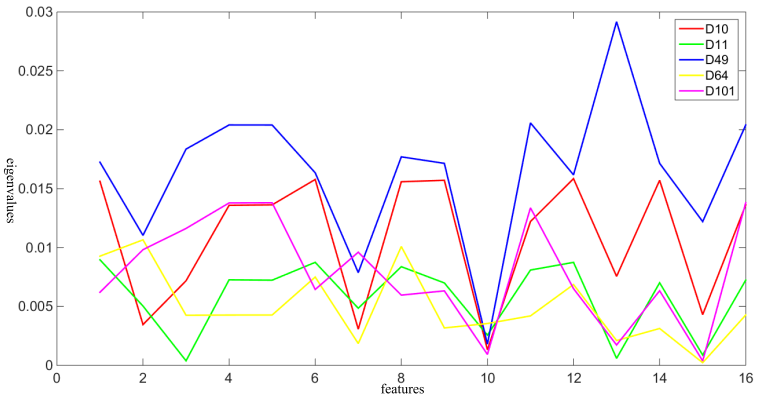


**Figure 7.** Difference of average eigenvalues between training and testing samples for Brodatz database.

## 5. Conclusions

This paper details a texture classification method based on the integrated optimization of the parameters and features of Gabor filter using HALO. Three public texture databases with different types of texture features were utilized for its evaluation. The experimental results were firstly compared with those of some commonly used swarm intelligence algorithms, such as PSO, DE, CS, and GWO. In general, it was demonstrated that swarm intelligence algorithms are well coded to solve parameter optimization and feature selection problems at the same time. Among them, HALO with hybrid binary-decimal coding has great optimization ability, and its fitness value is distinctly higher than that of the other algorithms. Thus, the proposed method is more appropriate for texture classification, and it is fast enough to meet real-time application needs. Moreover, for a more comprehensive comparison, Log-Gabor, DS-Gabor filter, and the only parameter optimization-based Gabor filter were also utilized. It was observed that the proposed texture classification method is robust, and the classification accuracy is satisfactory for multi-classification problems, especially those based on texture features. In sum, the multi-orientation and multiscale nature of Gabor filter can enhance the discrimination of texture features. Further, the disadvantage of high time complexity can be overcome to a great extent with HALO. The proposed texture classification method has an excellent balance between efficiency and accuracy, making it a good candidate to deal with practical applications. In the future, we will collect some geological texture samples and use them for the classification of more texture features.

**Author Contributions:** M.W. methodology; L.G. validation; X.H. formal analysis; Y.J. investigation; X.G. writing.

**Funding:** This work was funded by the National Key Research & Development Program of China under Grant No. 2017YFC1502406-03; the National Natural Science Foundation of China under Grant No. U1711266; and the Open Fund of State Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University under Grant No. 18R04.

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

## References

1. Haralick, R.M. Statistical and structural approaches to texture. *Proc. IEEE* **2005**, *67*, 786–804. [CrossRef]
2. Chen, W.; Li, X.; He, H.; Wang, L. Assessing different feature sets' effects on land cover classification in complex surface-mined landscapes by ZiYuan-3 satellite imagery. *Remote Sens.* **2017**, *10*, 23. [CrossRef]

3. Dong, Y.; Feng, J.; Liang, L.; Zheng, L.; Wu, Q. Multiscale sampling based texture image classification. *IEEE Signal Process. Lett.* **2017**, *24*, 614–618. [CrossRef]
4. Zhou, W.; Zhang, L.; Wang, K.; Chen, S.; Wang, G.; Liu, Z.; Liang, C. Malignancy characterization of hepatocellular carcinomas based on texture analysis of contrast-enhanced MR images. *J. Magn. Reson. Imaging* **2017**, *45*, 1476–1484. [CrossRef]
5. Liang, Y.; Li, Y.; Zhao, K; Meng, L. Object Tracking Algorithm based on Multi-channel Extraction of AHLBP Texture Features. In Proceedings of the International Conference on Advanced Mechatronic Systems (ICAMechS), Zhengzhou, China, 30 August–2 September 2018; pp. 332–336.
6. Banerjee, P.; Bhunia, A.K.; Bhattacharyya, A.; Roy, P.P.; Murala, S. Local Neighborhood Intensity Pattern—A new texture feature descriptor for image retrieval. *Expert Syst. Appl.* **2018**, *113*, 100–115. [CrossRef]
7. Islam, R.; Uddin, J.; Kim, J.M. Texture analysis based feature extraction using Gabor filter and SVD for reliable fault diagnosis of an induction motor. *Int. J. Inf. Technol. Manag.* **2018**, *17*, 20–32. [CrossRef]
8. Yuan, B.; Xia, B.; Zhang, D. Polarization Image Texture Feature Extraction Algorithm Based on CS-LBP Operator. *Procedia Comput. Sci.* **2018**, *131*, 295–301. [CrossRef]
9. Lloyd, K.; Rosin, P.L.; Marshall, D.; Moore, S.C. Detecting violent and abnormal crowd activity using temporal analysis of grey level co-occurrence matrix (GLCM)-based texture measures. *Mach. Vis. Appl.* **2017**, *28*, 361–371. [CrossRef]
10. Omar, M.; Khelifi, F.; Tahir, M.A. Detection and classification of retinal fundus images exudates using region based multiscale LBP texture approach. In Proceedings of the International Conference on Control, Decision and Information Technologies (CoDIT), St. Julian's, Malta, 6–8 April 2016; pp. 227–232.
11. Zuniga, A.G.; Florindo, J.B.; Bruno, O.M. Gabor wavelets combined with volumetric fractal dimension applied to texture analysis. *Pattern Recognit. Lett.* **2014**, *36*, 135–143. [CrossRef]
12. Wang, S.H.; Lin, C.Y.; Fu, J.T. Gender classification based on multi-scale and run-length features. *J. Electron. Sci. Technol.* **2017**, *15*, 251–257.
13. LLee, G.G.C.; Huang, C.H.; Chen, C.F.R.; Wang, T.P. Complexity-Aware Gabor Filter Bank Architecture Using Principal Component Analysis. *J. Signal Process. Syst.* **2017**, *89*, 431–444.
14. Li, C.; Huang, Y.; Zhu, L. Color texture image retrieval based on Gaussian copula models of Gabor wavelets. *Pattern Recognit.* **2017**, *64*, 118–129. [CrossRef]
15. Younesi, A.; Amirani, M.C. Gabor filter and texture based features for palmprint recognition. *Procedia Comput. Sci.* **2017**, *108*, 2488–2495. [CrossRef]
16. Huang, W.; Zhu, W.; Ma, C.; Guo, Y.; Chen, C. Identification of group-housed pigs based on Gabor and Local Binary Pattern features. *Biosyst. Eng.* **2018**, *166*, 90–100. [CrossRef]
17. Lu, R.S.; Dennison, E.; Denison, H.; Cooper, C.; Taylor, M.; Bottema, M.J. Texture analysis based on Gabor filters improves the estimate of bone fracture risk from DXA images. *Comput. Methods Biomech. Biomed. Eng. Imaging Vis.* **2018**, *6*, 453–464. [CrossRef]
18. Kim, N.C.; So, H.J. Directional statistical Gabor features for texture classification. *Pattern Recognit. Lett.* **2018**, *112*, 18–26. [CrossRef]
19. Khan, S.; Hussain, M.; Aboalsamh, H.; Mathkour, H.; Bebis, G.; Zakariah, M. Optimized Gabor features for mass classification in mammography. *Appl. Soft Comput.* **2016**, *44*, 267–280. [CrossRef]
20. Khan, S.; Khan, A.; Maqsood, M.; Aadil, F.; Ghazanfar, M.A. Optimized gabor feature extraction for mass classification using cuckoo search for big data e-healthcare. *J. Grid Comput.* **2018**, *8*, 1–16. [CrossRef]
21. Tong, L.; Wong, W.K.; Kwong, C.K. Differential evolution-based optimal Gabor filter model for fabric inspection. *Neurocomputing* **2016**, *173*, 1386–1401. [CrossRef]
22. Murugappan, V.; Sabeenian R.S. Texture based medical image classification by using multi-scale Gabor rotation-invariant local binary pattern (MGRLBP). *Cluster Comput.* **2017**, *12*, 1–14. [CrossRef]
23. Shen, L.; Zhu, Z.; Jia, S.; Zhu, J.; Sun, Y. Discriminative Gabor feature selection for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2013**, *10*, 29–33. [CrossRef]
24. Hussain, M. False positive reduction using Gabor feature subset selection. In Proceedings of the International Conference on Information Science and Applications (ICISA), Suwon, South Korea, 24–26 June 2013; pp. 1–5.
25. Mirjalili, S. The ant lion optimizer. *Adv. Eng. Softw.* **2015**, *83*, 80–98. [CrossRef]
26. Saikia, L.C.; Sinha, N. Automatic generation control of a multi-area system using ant lion optimizer algorithm based PID plus second order derivative controller. *Int. J. Electr. Power Energy Syst.* **2016**, *80*, 52–63.

27. Mouassa, S.; Bouktir, T.; Salhi, A. Ant lion optimizer for solving optimal reactive power dispatch problem in power systems. *Eng. Sci. Technol. Int. J.* **2017**, *20*, 885–895. [CrossRef]
28. Petrović, M.; Petronijević, J.; Mitić, M.; Vuković, N.; Miljković, Z.; Babić, B. The ant lion optimization algorithm for integrated process planning and scheduling. *Appl. Mech. Mater. Trans Tech Publ.* **2016**, *834*, 187–192. [CrossRef]
29. Yao, P.; Wang, H. Dynamic Adaptive Ant Lion Optimizer applied to route planning for unmanned aerial vehicle. *Soft Comput.* **2017**, *21*, 5475–5488. [CrossRef]
30. Mafarja, M.M.; Mirjalili, S. Hybrid binary ant lion optimizer with rough set and approximate entropy reducts for feature selection. *Soft Comput.* **2018**, *6*, 1–17. [CrossRef]
31. Shen, F.; Zhou, X.; Yang, Y.; Song, J.; Shen, H.T.; Tao, D. A fast optimization method for general binary code learning. *IEEE Trans. Image Process.* **2016**, *25*, 5610–5621. [CrossRef]
32. Song, X.; Liu, F.; Zhang, Z.; Yang, C.; Luo, X.; Chen, L. 2D Gabor filters-based steganalysis of content-adaptive JPEG steganography. *Multimedia Tools Appl.* **2017**, *76*, 26391–26419. [CrossRef]
33. Kong. R.; Zhang, B. Design of Gabor filters' parameter. *Control Decis.* **2012**, *27*, 1277–1280.
34. Lu, J.; Liu, F.; Luo, X. Selection of image features for steganalysis based on the Fisher criterion. *Digit. Investig.* **2014**, *11*, 57–66. [CrossRef]
35. Sheppard, JM.; Young, W.B. Agility literature review: Classifications, training and testing. *J. Sports Sci.* **2006**, *24*, 919–932. [CrossRef]
36. Shen, B.; Wang, Z.; Huang, T. Stabilization for sampled-data systems under noisy sampling interval. *Automatica* **2016**, *63*, 162–166. [CrossRef]
37. Textures.com, CGT Database. 26 September 2014. Available online: https://www.textures.com/ (accessed on 25 February 2019).
38. Kylberg, G. Kylberg Database. 1 September 2011. Available online: http://www.cb.uu.se/gustaf (accessed on 25 February 2019).
39. Picard, R.W.; Kabir, T.; Liu, F. Real-time recognition with the entire Brodatz texture database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New York, NY, USA, 15–17 June 1993; pp. 638–639.
40. Kenney, J. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; pp.1942–1948.
41. Storn, R; Price, K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]
42. Yang, X.S.; Deb, S. Cuckoo search via Lévy flights. In Proceedings of the World Congress on Nature & Biologically Inspired Computing (NaBIC), Coimbatore, India, 9–11 December 2009; pp. 210–214.
43. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
44. Nunes, F.G.; Pádua, L.C. A local feature descriptor based on Log-Gabor filters for key point matching in multispectral images. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1850–1854. [CrossRef]