Symmetry **2015**, *7*, 1734-1750; doi:10.3390/sym7041734

OPEN ACCESS

symmetry

ISSN 2073-8994

www.mdpi.com/journal/symmetry

Article

Fault Detection Based on Multi-Scale Local Binary Patterns Operator and Improved Teaching-Learning-Based Optimization Algorithm

Hongjian Zhang *, Ping He and Xudong Yang

Department of Control Science and Engineering, Harbin Institute of Technology, Harbin 150001, China; E-Mails: heping@hit.edu.cn (P.H.); yxdm@hit.edu.cn (X.Y.)

* Author to whom correspondence should be addressed; E-Mail: zhanghongjianhit@163.com; Tel.: +86-137-6683-7184.

Academic Editor: Sergei D. Odintsov

Received: 30 July 2015 / Accepted: 21 September 2015 / Published: 28 September 2015

Abstract: Aiming to effectively recognize train center plate bolt loss faults, this paper presents an improved fault detection method. A multi-scale local binary pattern operator containing the local texture information of different radii is designed to extract more efficient discrimination information. An improved teaching-learning-based optimization algorithm is established to optimize the classification results in the decision level. Two new phases including the worst recombination phase and the cuckoo search phase are incorporated to improve the diversity of the population and enhance the exploration. In the worst recombination phase, the worst solution is updated by a crossover recombination operation to prevent the premature convergence. The cuckoo search phase is adopted to escape the local optima. Experimental results indicate that the recognition accuracy is up to 98.9% which strongly demonstrates the effectiveness and reliability of the proposed detection method.

Keywords: fault detection; train center plate bolts loss; local binary patterns; teaching-learning-based optimization algorithm

1. Introduction

Fault detection of train center plate bolt loss (TCPBL) is an important part of the trouble of the moving freight car detection system (TFDS), which is highly demanded by the China Ministry of Railways now [1].

TCPBL has a high incidence which may lead to the train derailment if not found in time [2]. With trackside cameras photographing the key parts of the moving freight cars, four workers with more than five years' experience will check out all these photos to judge whether there are faults or not [3]. Our goal is to accomplish the same fault detection automatically by computer so that the efficiency of train safety inspection will be greatly improved. In TFDS, the status of freight cars and photography conditions vary significantly at different inspection stations such as illumination, weather, freight car speed, and camera vibration, *etc.* The photos can have the problems of blur, poor illumination, excess exposure, and occlusion. In recent years, only a few studies have been conducted with traditional feature extraction methods, such as the gray averages of local domain [4], the binary maps [5], and the Haar-like features [6]. However these methods tend to suffer when some factors (such as angle, illumination, and occlusion, *etc.*) change obviously. Therefore, the approach to a robust fault detection of TCPBL problem is researched.

Until recently, the local binary patterns (LBP) histogram is a simple yet efficient operator to extract the local texture information of images [7]. It was firstly used to extract features of human face images [8] and then widely applied on solving image recognition problems, such as automatic eye localization [9], fragile watermarking [10], fingerprint minutiae matching [11] and scene categorization [12]. Zhao *et al.* [13] combined the local statistic features of Gabor wavelets with the LBP features as the final feature vectors and obtained a good result in the person-independent facial expression recognition. Gabor wavelets have been widely used to establish new representations of different scales and orientations in the spatial domain [14–16] which is beneficial to extract the image texture features. Nouyed *et al.* [17] pointed out that features of different Gabor channels had different contributions to the final classification task and reported a weighted voting face recognition method. Gao *et al.* [18] proposed an adaptively weighted local multi-channel Gabor filters (AWMGF) method, which determined the weights of local Gabor magnitude maps by the contribution of each sub-pattern to the classification. Zhu *et al.* [19] introduced the adaptive boosting (AdaBoost) algorithm to select the important Gabor features in the facial expression recognition. However, the optimal weights of different channels are difficult to calculate in theory, powerful optimization method is needed to search the global optimal weights in the multidimensional space. Rao *et al.* developed an effective teaching-learning-based optimization (TLBO) algorithm which performed well on the mechanical design problems [20], the continuous non-linear large scale problems [21] and the heat exchangers problems [22]. Togan *et al.* [23] presented a design procedure employing the TLBO to the discrete optimization of planar steel frames. Yu *et al.* [24] applied TLBO on several numerical and engineering optimization problems and proved that TLBO is more powerful than the improved bee algorithm (IBA) [25], the hybrid particle swarm optimization with differential evolution (PSO-DE) [26], the modified differential evolution algorithm (COMDE) [27], the g-best guided artificial bee colony (GABC) [28] and the upgraded artificial bee colony (UABC) [29] algorithms.

Previous studies have proven the efficiency of LBP features and TLBO algorithm on engineering applications, however little work has been conducted to investigate their effectiveness on the TCPBL problem. Therefore, an improved fault detection method is proposed to offer a reliable solution to the TCPBL problem against different photographing conditions. A multi-scale local binary patterns (MLBP) operator is designed to extract the texture features of new representations of different Gabor channels. With information of different scales and texture change trends of local areas coded, it can obtain more efficient texture information. An improved teaching-learning-based optimization (ITLBO) algorithm with the worst recombination phase and the cuckoo search phase is proposed to optimize the weight of the classification results of different channels. The experiment results verify the global search ability of the ITLBO and the effectiveness of the proposed detection method.

The rest of this article is organized as follows: Section 2 firstly introduces the structure of the proposed fault detection method and then gives a detailed description on the design of the MLBP operator and the ITLBO algorithm. Then two experimental results are given in Section 3: Compared with some state-of-the-art methods, the experimental results on the benchmark problems can evaluate the search capability of the ITLBO, and the experimental results on the field images database demonstrates the effectiveness of the proposed detection method. Finally, the conclusions of the present work and future plans are given in Section 4.

2. Fault Detection Based on MLBP and ITLBO

The framework of the proposed fault detection method is presented in Figure 1. Gabor wavelets transform is used to acquire the multi-channel representations from different scales and orientations. Then, the MLBP operator is adopted to extract the local texture features of each channel separately. The results of the support vector machine (SVM) classifiers of different channels are usually treated equally, however it will have a positive influence on the recognition task to weight each channel differently. Therefore, the ITLBO algorithm is established to optimize the weights of classification results of different channels with its powerful search capability in global-optimization. Finally, the detection results are acquired by the decision-level weight fusion.

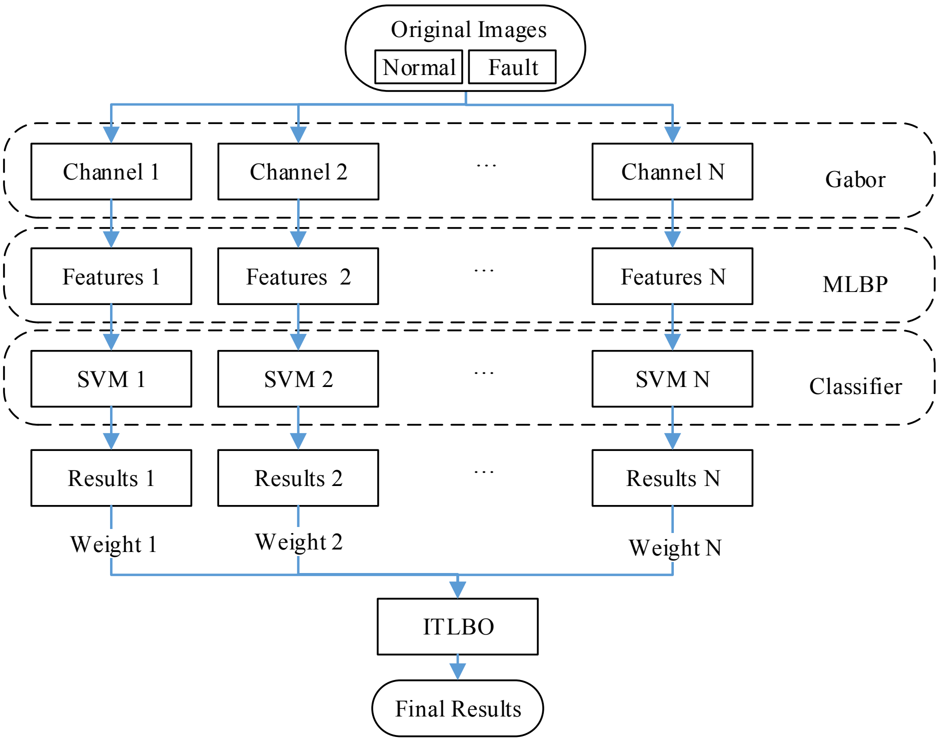


Figure 1. The framework of the proposed fault detection approach.

2.1. Multi-Scale LBP Operator

The LBP operator is a powerful local texture descriptor for image feature extraction, which is defined as:

$$LBP_{P,R} = \sum_{i=0}^{P\text{-}1} s\ (g_i - g_c)\cdot 2^i,\ \ s\ (g_i - g_c)\ =\ \begin{cases} 1 & ,g_i \geq g_c \\ 0 & ,g_i < g_c \end{cases} \tag{1}$$

where g_c is the gray value of the center point. $g_i (i = 0, \ldots, P-1)$, is the P number of neighborhood points on a circle of radius R with g_c as the center.

The LBP operator has been successfully used in the image recognition since it has good robustness on the influence of illumination, angle, and occlusion. However, it does not perform well when faced with severe illumination changes and it is too vague to describe the unique details contained in an image. Thus, an improved multi-scale LBP operator (MLBP) is designed to extract more distinctive information and improve the recognition performance. The center point is compared with multiple neighborhood points of different radii and the neighborhood points in the same radial directions are also compared with each other. In this article, two radii and eight radial directions are selected, which are similar to two LBP operators of different radii. The structure of the MLBP operator is shown in Figure 2.

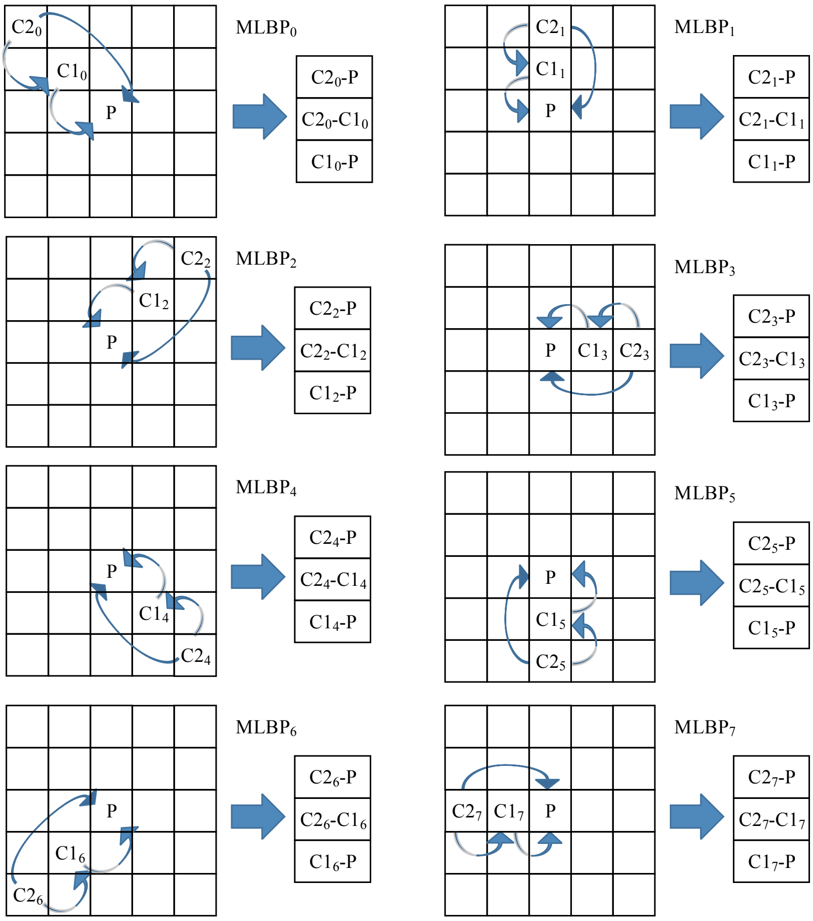
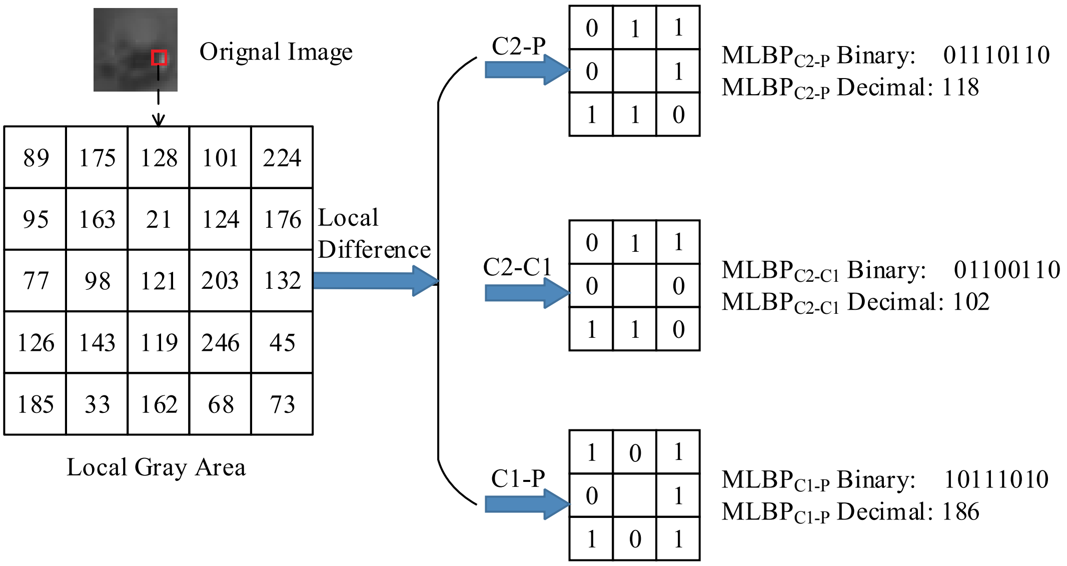


Figure 2. The MLBP operator model.

A region of 5 × 5 points is selected, which is composed of two squares with different radii but centered in the same point *P*. $C1_i (i = 0,1,L,7)$ is the neighborhood point on the inner square with the smaller radius $R1$ = 1 and $C2_i(i = 0,1,\ldots,7)$ is the neighborhood point on the outer square with the bigger radius $R2$ = 2. The pixel values of eight radial directions are compared with each other in accordance with the thick arrows indications. The 3 × 8 local difference values of three arrows in eight clockwise orientations from the center point compose the whole MLBP operator. Referring to the definition of the original LBP operator, only the important sign information is coded by:

$$MLBP_{diff} = \sum_{i=0}^{7} s(value_{diff}) \cdot 2^{i}, s(value_{diff}) = \begin{cases} 1 & value_{diff} > 0 \\ 0 & otherwise \end{cases} \quad (2)$$

where $diff \in (C1_i - P, C2_i - P, C2_i - C1_i)$, is the local gray difference value from the tail to the head of the arrows. Here $C1_i - P$ is the small scale descriptor of the inner square, which is similar to the $LBP_{8,1}$ operator; $C2_i - P$ is the big scale descriptor of the outer square, which is similar to the $LBP_{8,2}$ operator; $C2_i - C1_i$ is the supplementary explanation to describe the texture changes between the inner square and the outer square. By coding these three descriptors into the MLBP features, more comprehensive and distinguishable texture information can be obtained than using only one of them. Besides, the radii $R1$, $R2$ are changeable to obtain better performance. A complete MLBP model is shown in Figure 3.



89	175	128	101	224
95	163	21	124	176
77	98	121	203	132
126	143	119	246	45
185	33	162	68	73

0	1	1
0		1
1	1	0

0	1	1
0		0
1	1	0

1	0	1
0		1
1	0	1

Figure 3. A complete MLBP feature.

2.2. Improved TLBO Algorithm

TLBO is an interactive teaching-learning process-inspired algorithm based on the influence of a teacher on the achievements of learners in a class. Although the TLBO algorithm has a fast convergence speed, it may be trapped in the local optima on some difficult optimization problems.

2.2.1. Basic TLBO

TLBO is a population-based method. Learners in a classroom are considered as the population and different subjects taken by the learners are considered as different design variables of the optimization problems. Achievements of learners are analogous to the fitness values of the optimization problems. The one with the best fitness value in the entire population is considered as the teacher of the class. The basic TLBO algorithm can be divided into two parts: the teacher phase and the learner phase. The working of these two phases are explained as below.

Teacher Phase

During this phase the teacher tries to increase the mean achievement of the class towards his or her level in terms of knowledge. However the teacher can only increase the mean achievement of the class up to some extent depending on the learners' capability. Moreover, the achievements of the learners cannot increase equally as different learners possess different knowledge levels. This aspect is presented in mathematical form as follows:

Assume that at any iteration $i(i = 1,2,\ldots,G)$, there are D number of subjects (*i.e.*, design variables) offered to N number of learners (*i.e.*, population size). $X_{j,k,i}$ is the knowledge of the learner $k(k = 1,2,\ldots,N)$ in a particular subject $j(j = 1,2,\ldots,D)$, $M_{j,i}$ is the mean knowledge level of the learners in the subject j and $T_{j,i}$ is the knowledge of the teacher (*i.e.*, best solution) in the subject j. The learner k is updated according to the difference between the current and the new mean knowledge levels of each subject given by:

$$Difference\_Mean_{j,k,i} = r\left(M^{new} - T_F M_{j,i}\right) \quad (3)$$

where r is a uniformly distributed random number within the range [0,1]. As mentioned earlier, $T_{j,i}$ tries to move the mean knowledge level $M_{j,i}$ towards its own level, so now $T_{j,i}$ will be designated as the new mean knowledge level, then $M^{new} = T_{j,i}$. T_F is the teaching factor which decides the value of mean knowledge level to be changed and its value is decided by:

$$T_F = round[1 + rand\left(0,1\right)\{2-1\}] \quad (4)$$

where the value of T_F can be either 1 or 2 randomly with equal probability which means T_F is not a given input parameter of the algorithm.

$Difference\_Mean_{j,k,i}$ modifies the current learner k according to the following expression:

$$X_{j,k,i}^{new} = X_{j,k,i} + Difference\_Mean_{j,k,i} \quad (5)$$

where $X_{j,k,i}^{new}$ is the updated value of $X_{j,k,i}$ and is accepted if it gives a better fitness value than $X_{j,k,i}$.

Learner Phase

During this phase learners interact with other learners and increase their own knowledge if other learners have more useful knowledge than themselves. The learner phase is expressed as below:

At ith iteration, each learner is compared with another learner randomly. For learner $X_{P,i}$, learner $X_{Q,i}$ are randomly selected such that $X_{P,i} \neq X_{Q,i}$. In the case that if $X_{Q,i}$ is better than $X_{P,i}$, $X_{P,i}$ is moved toward $X_{Q,i}$. Otherwise it is moved away from $X_{Q,i}$.

$$X_{j,P,i}^{new} = \begin{cases} X_{j,P,i} + r\left(X_{j,P,i} - X_{j,Q,i}\right), & if\ fitness\left(X_{P,i}\right) bettern\ than\ fitness\left(X_{Q,i}\right) \\ X_{j,P,i} + r\left(X_{j,Q,i} - X_{j,P,i}\right), & else \end{cases} \quad (6)$$

where r is a random number in the range 0 to 1. $X_{j,P,i}^{new}$ is the updated value of $X_{j,P,i}$ and accepted if it gives a better fitness value than $X_{j,P,i}$.

2.2.2. Improved TLBO Algorithm

Two modifications are incorporated into the basic TLBO algorithm to improve its global search capability and avoid being trapped in the local optima.

Worst Recombination Phase

Rao and Patel have introduced the elite TLBO algorithm, in which, the worst learner is replaced by the elite learner after the learner phase [30]. If duplicate learners exist after the replacement, one randomly selected subject of the duplicate learner is modified into a random value. However the diversity of the population can be gradually lost. Meanwhile knowledge of all subjects of the worst learner is discarded during the replacement. However the worst learner does not mean his knowledge of all subjects is the worst. Actually, he can have useful knowledge in some subjects for the further search. Therefore, a new worst recombination phase is proposed to maintain the population diversity and enhance the exploration search. Similar to the crossover operator of differential evolution [31], the worst learner is replaced by the crossover recombination of the current worst learner and another learner randomly; thus useful knowledge of some subjects of the worst learner is still maintained in the new learner. The worst recombination phase is implemented after the learner phase and is explained as below:

At *i*th iteration, let the learner with the worst fitness value be $X_{P,i}$. For recombination, another learner $X_{Q,i}$ is randomly selected such that $X_{P,i} \neq X_{Q,i}$.

$$X_{j,P,i}^{new} = \begin{cases} X_{j,P,i}, & r_j \leq CP \ or \ j = j_{rand} \\ X_{j,Q,i}, & otherwise \end{cases} \tag{7}$$

where $X_{j,P,i}^{new}$ is the recombination of learner $X_{P,i}$ and $X_{Q,i}$ in the subject *j*. $r_j \in [0,1]$, is a uniformly distributed random number generated for each subject. $CP \in [0,1]$, is the crossover probability constant to decide the distribution proportion of two learners in all subjects. $j_{rand} \in [1, D]$, is a random selected subject to ensure that $X_{P,i}^{new}$ inherits at least one subject of $X_{P,i}$. $X_{P,i}^{new}$ is accepted if it gives a better fitness value than $X_{P,i}$.

Cuckoo Search Phase

In the basic TLBO algorithm, learners get improved by moving towards other solutions with higher fitness values through the input of their teacher and the interaction with other learners. However, it lacks the jumping out operation when the algorithm is trapped in a local optimum. Cuckoo search is used to prevent the search from running into a local optimum and overcome the shortcomings of the basic TLBO, which is inspired by the breeding behavior of cuckoos laying their eggs in the nests of other species and the Lévy flights of birds [32]. The Lévy flight is a random walk in which the step-lengths have a heavy-tailed distribution with the directions of the steps being isotropic and random, so that the population can move in random directions. Here, the randomicity of the cuckoo search is used to avoid being trapped in the local optima.

The learners of a class are divided into two groups according to their achievements (*i.e.*, fitness values). Then, multilevel knowledge can be taught in accordance with the capability of different groups of learners. The elite learners are selected for gentle teaching-learning process while the other learners become the inputs of cuckoo search phase for severe training.

2.2.3. Procedure of the ITLBO

With the above design, the flowchart of the proposed ITLBO algorithm is illustrated in Figure 4.

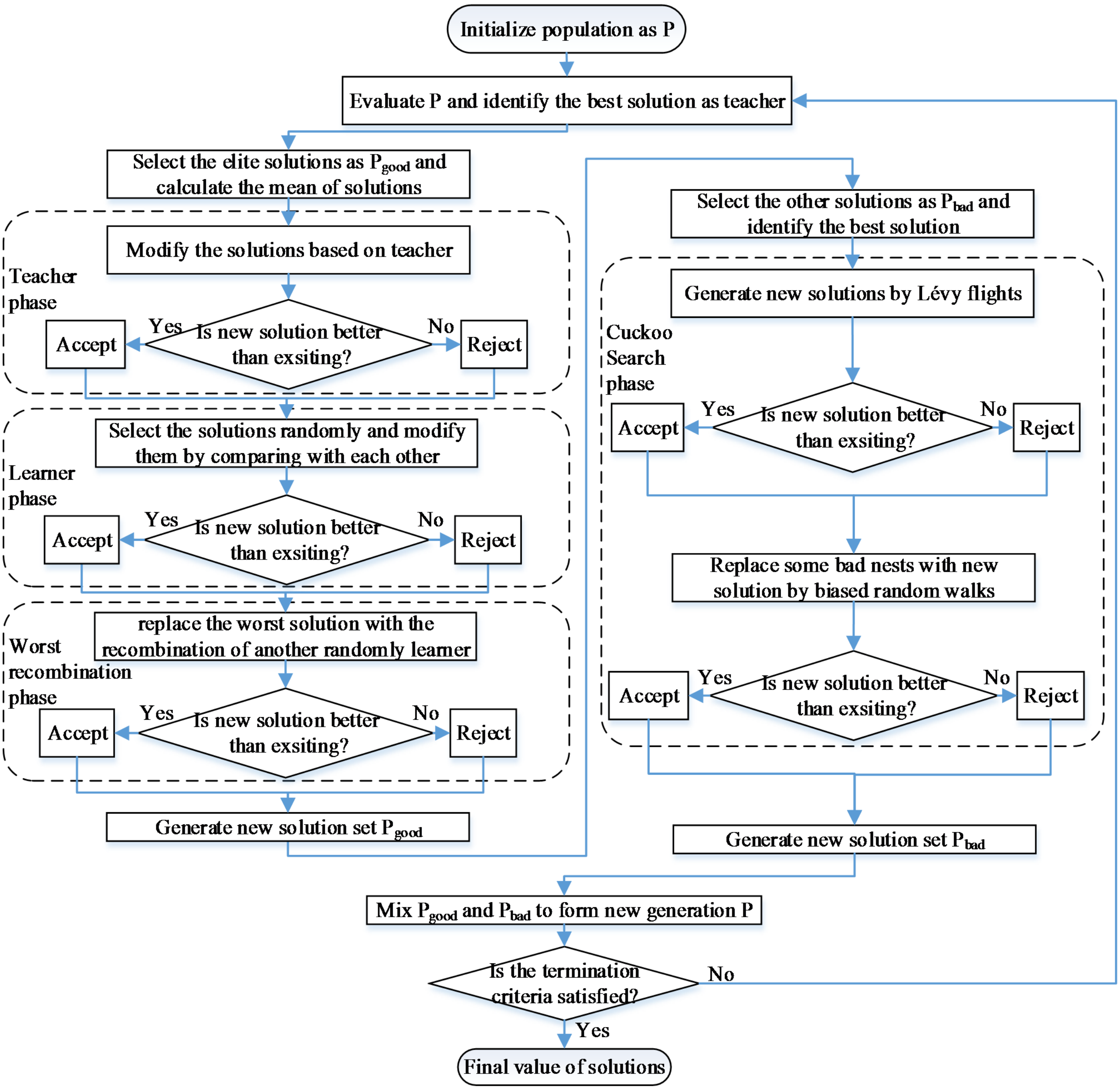

Initialize population as P
Evaluate P and identify the best solution as teacher
Select the elite solutions as P_{good} and calculate the mean of solutions
Teacher phase
Modify the solutions based on teacher
Accept
Yes
Is new solution better than exsiting?
No
Reject
Learner phase
Select the solutions randomly and modify them by comparing with each other
Accept
Yes
Is new solution better than exsiting?
No
Reject
Worst recombination phase
replace the worst solution with the recombination of another randomly learner
Accept
Yes
Is new solution better than exsiting?
No
Reject
Generate new solution set P_{good}
Select the other solutions as P_{bad} and identify the best solution
Cuckoo Search phase
Generate new solutions by Lévy flights
Accept
Yes
Is new solution better than exsiting?
No
Reject
Replace some bad nests with new solution by biased random walks
Accept
Yes
Is new solution better than exsiting?
No
Reject
Generate new solution set P_{bad}
Mix P_{good} and P_{bad} to form new generation P
Is the termination criteria satisfied?
No
Yes
Final value of solutions

Figure 4. The flowchart of ITLBO.

First, it generates *N* learners randomly as an initial population *P*. Then the elite learners with higher achievements are selected as the inputs of the teacher phase, learner phase and worst recombination phase. Next, the other learners become the inputs of the cuckoo search phase, in which all solutions (except for the best one) are updated by the Lévy flights mutation and some bad solutions are replaced with new solutions by the biased random walks. Since multiple operators including teaching-learning, crossover recombination, and Lévy mutation is used, exploration and exploitation can be balanced and a better global search ability with faster convergence speed can be obtained.

2.2.4. Time Complexity Analysis

The ITLBO algorithm can be mainly divided into the initialization, the elite learners' selection, the teacher phase, the learner phase, the worst recombination phase, and the cuckoo search phase. The time complexity of each procedure is analyzed as below:

The initialization of N number of learners on D number of subjects is performed with the time complexity $O[N \times D]$. The elite learners' selection composed of ranking the learners costs the time complexity $O[N \times \log N]$. Both the teach phase and the learner phase have the same time complexity $O[N \times D]$. The worst recombination phase: the crossover recombination operation costs the time complexity $O[D]$. The cuckoo search phases: both the Lévy flights mutation and the bad nests replacement are performed with the time complexity $O[N \times D]$, so the total time complexity of this phase is still $O[N \times D]$.

Therefore, the total time complexity of the proposed ITLBO algorithm is $O[N \times D] + O[N \times \log N] + O[N \times D] + O[N \times D] + O[D] + O[N \times D]$, which can be simplified as $O[N \times D] + O[N \times \log N]$. Considering that the population size N is limited to a constant value and the D of optimization problems are usually much larger than $\log N$, so the total time complexity of the ITLBO is considered as $O[N \times D]$. Taking the iteration number G into consideration, the final computational complexity of the TLBO is $O[G \times N \times D]$, which is not large and is the same as that of the basic TLBO. Besides, the total number of function evaluations in the ITLBO algorithm is still $2 \times G \times N$, which is also the same as that of the basic TLBO.

3. Experiments, Results, and Discussions

3.1. Experiments on Benchmark Functions

Eight standard benchmark functions with different characters shown in Table 1 were considered to evaluate the performance of the proposed ITLBO algorithm. The numerical results of the ITLBO were compared with those of the basic TLBO and other state-of-the-art optimization algorithms, such as the TLBO algorithm with Lévy mutation in the learner phase (LTLBO) [33], the genetic algorithm with multi-parent crossover (GAMPC) [34] and the self-regulation particle swarm optimization algorithm (SRPSO) [35]. The "D" refers to the dimension of the optimization problems. The "Range" is the lower and upper bounds of the search spaces. The "$f_{\min}$" is the theoretical global minimum value for these problems. All algorithms have been coded using Matlab R2014a, and have been run on a PC with a 2.6 GHz processor, 4GB RAM, and Windows 10.

Table 1. Test functions considered in the experiment.

Function	Formula	Range	$f_{\min}$
Sphere	$f_1(x)=\sum_{i=1}^{D} x_i^2$	[−100, 100]	0
Rosenbrock	$f_2(x)=\sum_{i=1}^{D-1} 100(x_{i+1}-x_i^2)^2+(1-x_i)^2$	[−2.048, 2.048]	0
Schwefel P1.2	$f_3(x)=\sum_{i=1}^{D}(\sum_{j=1}^{i} x_j)^2$	[−100, 100]	0
Schwefel P2.22	$f_4(x)=\sum_{i=1}^{D}\lvert x_i\rvert+\prod_{i=1}^{D}\lvert x_i\rvert$	[−10, 10]	0
Schwefel P2.21	$f_5(x)=\max\{\lvert x_i\rvert, 1\le i\le D\}$	[−100, 100]	0
Rastrigin	$f_6(x)=\sum_{i=1}^{D}\left(x_i^2-10\cos(2\pi x_i)+10\right)$	[−5.12, 5.12]	0
Griewank	$f_7(x)=\sum_{i=1}^{D}\frac{x_i^2}{4000}-\prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}})+1$	[−600, 600]	0
Ackley	$f_8(x)=20-20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}\right)-\exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right)+e$	[−32.768, 32.768]	0

To decrease the statistical errors, each function was carried out with random seeds for 25 independent runs. For a fair comparison, the maximum function evaluations of these five algorithms were set to 3000 and all test functions were simulated in 10, 30, and 50 dimensions separately. The population size of each algorithm was set to 20. As the TLBO is a parameter-less algorithm, no other parameter is required for the working of the TLBO. The crossover probability constant *CP* was set to 0.5 for the ITLBO algorithm. The scaling factor $\gamma=1$ and the shape factor $\alpha=0.8$ were set for the learner phase with Lévy mutation of the LTLBO. The parameters of GAMPC are set to: $m=10, cr=100\%, \rho=[0.05, 0.1, 0.1]$. As for the SRPSO, the acceleration rate is set as $\eta=1$ and the initial and final values of inertia weight are set to: $\omega_I=1.05, \omega_F=0.55$.

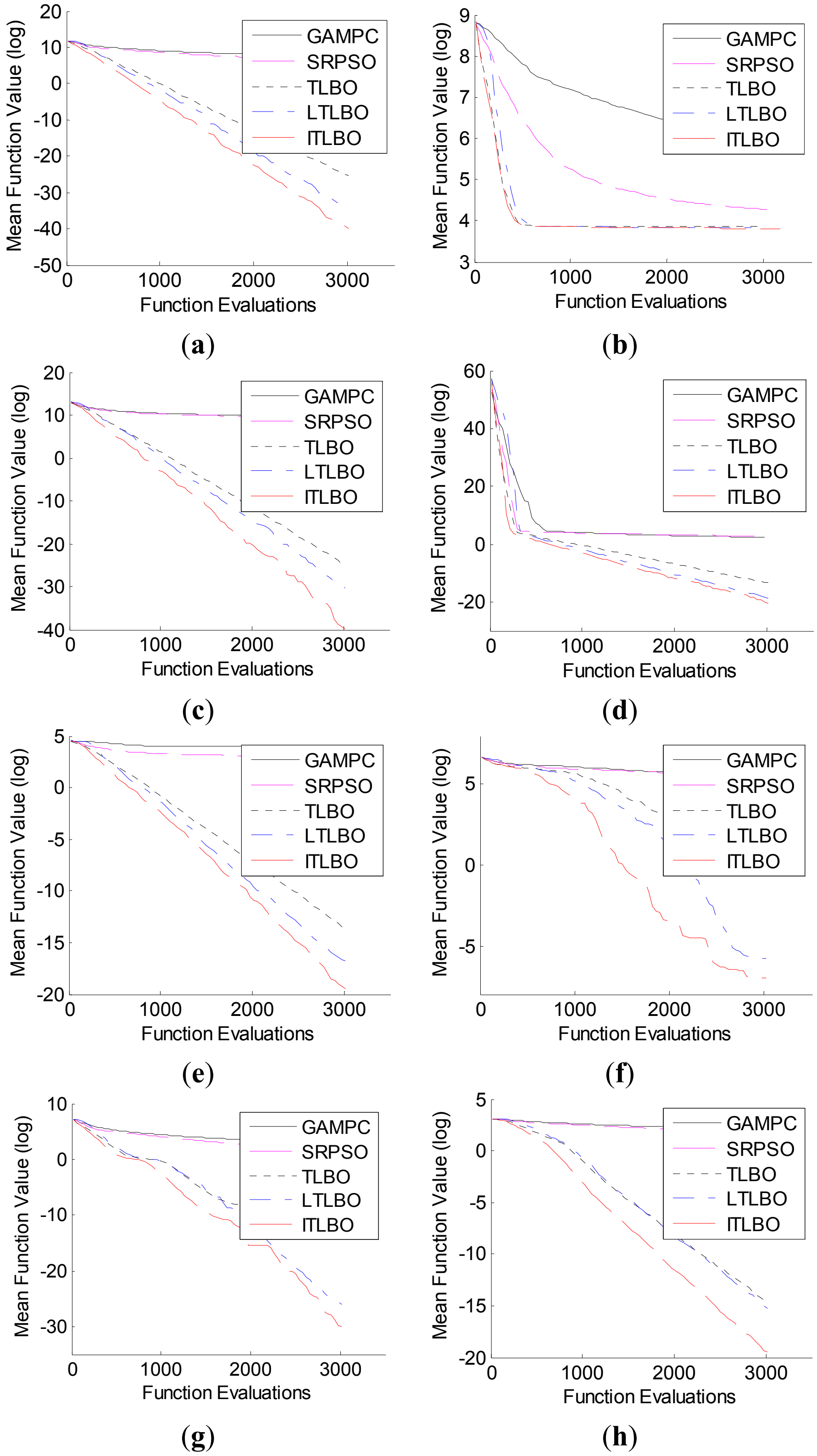
The comparative results of eight benchmark functions for five algorithms are presented in Table 2 in the form of the mean results (M) and the standard deviation (SD). The notations F, A, and D denote function, algorithm, and dimension, respectively. The boldface values given in the table indicate the best results among these five algorithms.

The results show that the TLBO-based algorithms often obtain better solutions than the GAMP and SRPSO algorithms, which indicates the strong search capability of the TLBO. Especially, the ITLBO obtains all the best results of seven functions in 10, 30, and 50 dimensions. Although the SRPSO acquires the best mean results for the Rosenbrock function (D = 10, 30), the ITLBO outperforms the other four algorithms under a higher dimension (D = 50). The performance of ITLBO is greatly improved than that of TLBO and LTLBO, which proves the effectiveness of the worst recombination phase and the cuckoo search phase. The performance of these algorithms commonly decreases when the dimension is higher but the performance of TLBO-based algorithms on the Griewank and Rastrigin functions is just on the contrary. Črepinšek *et al.* [36] pointed out that, the fitness-distance correlations of some functions increase with the dimensions. For example, for Griewank (D = 2) the fitness-distance correlation is 0.45, while for (D = 50) it is 0.99. When the fitness-distance correlation increases with the dimension, such problems are actually easier to solve for TLBO-based algorithms. This explained the results on the Griewank and Rastrigin functions are better under higher dimensions.

Table 2. Comparison results of ITLBO and state-of-art algorithms on eight test functions of 10, 30, and 50 dimensions with 3000 function evaluations.

F	A	D = 10		D = 30		D = 50	
		M	SD	M	SD	M	SD
Sphere	GAMPC	1.61×10^{-6}	7.56×10^{-6}	1.93×10^{2}	3.52×10^{2}	1.54×10^{3}	8.05×10^{2}
	SRPSO	9.95×10^{-1}	5.85×10^{-1}	4.93×10	1.33×10	3.73×10^{2}	8.83×10
	TLBO	7.16×10^{-13}	1.05×10^{-12}	1.76×10^{-12}	2.23×10^{-12}	2.49×10^{-12}	3.03×10^{-12}
	LTLBO	1.75×10^{-17}	5.54×10^{-17}	4.92×10^{-16}	1.02×10^{-15}	1.53×10^{-14}	5.71×10^{-14}
	ITLBO	**6.92×10^{-21}**	**2.37×10^{-20}**	**3.74×10^{-19}**	**1.39×10^{-18}**	**7.65×10^{-18}**	**3.32×10^{-17}**
Rosenbrock	GAMPC	5.10	1.85	4.68×10	1.72×10	1.44×10^{2}	5.50×10
	SRPSO	**4.20**	6.71×10	**2.34×10**	3.52	4.82×10	3.86
	TLBO	7.47	5.54×10^{-1}	2.74×10	7.48×10^{-1}	4.71×10	9.65×10^{-1}
	LTLBO	7.66	**3.91×10^{-1}**	2.67×10	6.36×10^{-1}	4.64×10	1.11
	ITLBO	6.91	4.79×10^{-1}	2.58×10	**5.68×10^{-1}**	**4.54×10**	**7.93×10^{-1}**
Schwefel P1.2	GAMPC	1.63	5.81	2.33×10^{3}	1.40×10^{3}	1.62×10^{4}	5.78×10^{3}
	SRPSO	11.49	4.91	3.13×10^{3}	9.46×10^{2}	1.31×10^{4}	4.87×10^{3}
	TLBO	1.07×10^{-12}	2.98×10^{-12}	5.91×10^{-12}	1.01×10^{-11}	1.07×10^{-11}	1.58×10^{-11}
	LTLBO	7.76×10^{-17}	1.56×10^{-16}	9.39×10^{-15}	2.85×10^{-14}	4.75×10^{-13}	2.29×10^{-12}
	ITLBO	**4.41×10^{-19}**	**1.65×10^{-18}**	**5.37×10^{-19}**	**1.33×10^{-18}**	**6.62×10^{-18}**	**2.63×10^{-17}**
Schwefel P2.22	GAMPC	4.00×10^{-7}	1.13×10^{-6}	5.53×10^{-1}	5.58×10^{-1}	1.00×10	4.63
	SRPSO	2.20×10^{-1}	5.43×10^{-2}	3.90	1.60	1.54×10	3.79
	TLBO	2.61×10^{-7}	1.89×10^{-7}	7.83×10^{-7}	4.99×10^{-7}	1.57×10^{-16}	1.64×10^{-6}
	LTLBO	1.41×10^{-9}	2.20×10^{-9}	5.57×10^{-9}	7.18×10^{-9}	1.25×10^{-8}	1.64×10^{-8}
	ITLBO	**2.26×10^{-11}**	**3.85×10^{-11}**	**1.33×10^{-10}**	**1.75×10^{-10}**	**4.01×10^{-10}**	**8.51×10^{-10}**
Schwefel P2.21	GAMPC	5.64	5.83	4.11×10	1.08×10	5.80×10	1.06×10
	SRPSO	9.31×10^{-1}	2.30×10^{-1}	1.03×10	1.98	1.89×10	2.98
	TLBO	8.16×10^{-7}	5.79×10^{-7}	1.43×10^{-6}	1.62×10^{-6}	1.28×10^{-6}	1.19×10^{-6}
	LTLBO	1.96×10^{-9}	2.34×10^{-9}	1.03×10^{-8}	1.70×10^{-8}	2.61×10^{-8}	3.74×10^{-8}
	ITLBO	**1.08×10^{-10}**	**1.24×10^{-10}**	**1.26×10^{-9}**	**2.32×10^{-9}**	**2.15×10^{-9}**	**5.36×10^{-9}**
Rastrigin	GAMPC	8.22	6.33	8.29×10	3.74×10	1.75×10^{2}	5.65×10
	SRPSO	1.42×10	5.35	9.16×10	2.85×10	1.93×10^{2}	2.74×10
	TLBO	1.34×10	1.53×10	3.14×10	7.00×10	2.62×10^{-1}	1.31
	LTLBO	1.84	6.59	2.59×10^{-4}	8.79×10^{-4}	4.95×10^{-5}	2.05×10^{-4}
	ITLBO	**5.59×10^{-1}**	**2.79**	**8.04×10^{-7}**	**3.99×10^{-6}**	**4.80×10^{-12}**	**1.23×10^{-11}**
Griewank	GAMPC	1.47×10^{-1}	1.77×10^{-1}	2.16	2.93	1.68×10	1.25×10
	SRPSO	7.86×10^{-1}	1.81×10^{-1}	1.49	1.17×10^{-1}	4.39	8.24×10^{-1}
	TLBO	3.59×10^{-2}	1.17×10^{-1}	8.84×10^{-12}	1.14×10^{-11}	8.08×10^{-12}	1.61×10^{-11}
	LTLBO	2.49×10^{-10}	1.15×10^{-10}	1.45×10^{-12}	1.92×10^{-12}	4.63×10^{-12}	1.09×10^{-11}
	ITLBO	**5.15×10^{-11}**	**2.56×10^{-10}**	**3.10×10^{-16}**	**6.13×10^{-16}**	**2.88×10^{-16}**	**5.66×10^{-16}**
Ackley	GAMPC	9.24×10^{-2}	3.19×10^{-1}	4.06	1.86	8.42	1.73
	SRPSO	9.26×10^{-2}	2.96×10^{-1}	3.53	2.72×10^{-1}	5.28	6.19×10^{-1}
	TLBO	2.18×10^{-7}	1.28×10^{-7}	3.77×10^{-7}	3.18×10^{-7}	4.11×10^{-7}	2.55×10^{-7}
	LTLBO	3.23×10^{-8}	7.72×10^{-8}	1.76×10^{-7}	3.53×10^{-7}	2.19×10^{-7}	2.25×10^{-7}
	ITLBO	**1.26×10^{-9}**	**1.40×10^{-9}**	**1.26×10^{-9}**	**9.76×10^{-10}**	**3.34×10^{-9}**	**6.56×10^{-9}**

Figure 5 was plotted between the mean function values in log scale and the function evaluations over 25 runs. Five algorithms for eight benchmark functions of 50 dimensions were selected to evaluate the convergence speed of the GAMPC, SRPSO, TLBO, LTLBO, and ITLBO algorithms. It makes clear that the ITLBO has much faster convergence speed than other four algorithms.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**) (**g**) (**h**)

Figure 5. Convergence graphs of eight 50-dimensional functions for GAMPC, SRPSO, TLBO, LTLBO and ITLBO: (**a**) Sphere; (**b**) Rosenbrock; (**c**) Schwefel P1.2; (**d**) Schwefel P2.22; (**e**) Schwefel P2.21; (**f**) Rastrigin; (**g**) Griewank; (**h**) Ackley.

All the results indicate that the ITLBO has a better convergence speed and stronger global search capability compared with the GAMPC, SRPSO, TLBO, and LTLBO algorithms. With the help of the worst recombination phase and the cuckoo search phase, ITLBO can get a good population diversity and avoid hovering around the local optimum solutions.

3.2. Application on Fault Detection of TCPBL Problem

3.2.1. Experiment Database

This article mainly studies on the recognition method of bolts region images. Therefore, the experiment database is composed of bolts region images from the segmentation of region of interest of the field images with fault detection manually checked, which contains 500 normal images and 500 fault images. Each image has a resolution of 32 × 32 pixels. Some of them are shown in Figure 6. Because these images are photographed in the process of train movement by the high speed trackside cameras, they have the problems of blur, poor illumination, excess exposure, and occlusion.

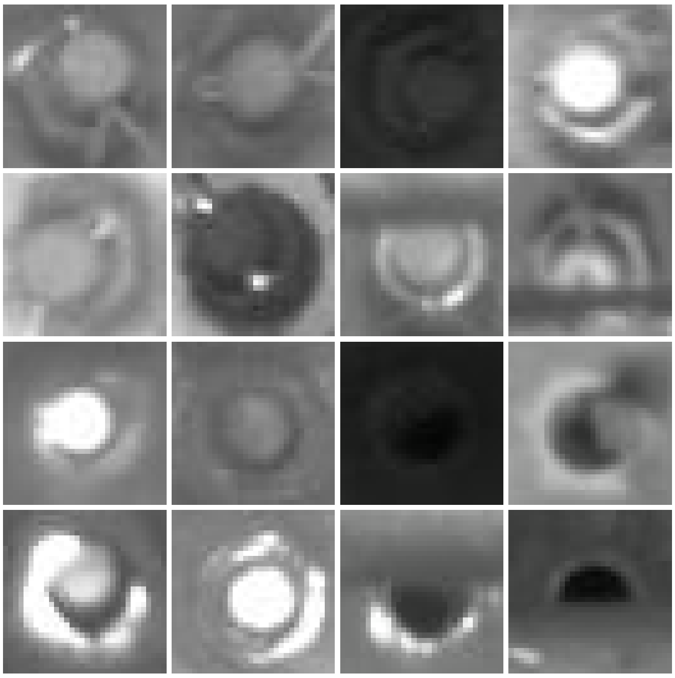


Figure 6. Some samples in the bolts region images database.

3.2.2. Recognition Results and Discussions

In this article, Gabor wavelets transform was used at five scales and eight orientations; thus, the dimension of this optimization problem was 40. In the process of classification, the Leave-One-Out Cross Validation (LOO-CV) was adopted to reduce the influence of stochastic factors and improve the reliability of the classification results. To evaluate the performance of this approach, several different optimization methods with the LBP and MLBP operators are compared and the best (B), mean (M) recognition rates (%) and the standard deviation (SD) over 25 independent runs are given in Table 3.

Table 3. Comparison results of fault detection with different methods.

Weight Optimization Method	LBP			MLBP		
	B	**M**	**SD**	**B**	**M**	**SD**
No Optimization	–	94.60	–	–	96.30	–
AWMGF	–	94.80	–	–	96.30	–
AdaBoost	–	95.10	–	–	96.50	–
GAMPC	96.60	96.13	0.38	98.20	97.03	0.27
SRPSO	96.80	96.01	0.27	98.00	97.24	0.25
TLBO	97.00	96.36	0.23	98.20	97.57	0.31
LTLBO	97.20	96.40	0.36	98.40	97.68	0.26
ITLBO	97.60	96.96	0.26	98.90	98.56	0.18

The results show that the MLBP operator has much higher accuracy than LBP operator because it can maintain more distinguishable texture information including not only the local difference information of two different scales but also the texture changes between different scales. Furthermore, with proper weights assigned to the classification results of different Gabor channels, the recognition rates are increased. Moreover, as for this weight optimization problem, the proposed ITLBO algorithm is more efficient than other methods, such as GAMPC, SRPSO, TLBO, and LTLBO. With the help of MLBP and ITLBO, the proposed fault detection method is reliable and efficient in practice.

4. Conclusions

In this article, an improved fault detection method based on the MLBP operator and ITLBO algorithm is successfully applied to the TCPBL problem of TFDS, which is a new application area of pattern recognition. This study reveals that the MLBP operator can obtain more comprehensive discrimination information than LBP, which maintains both the local textures and the texture changes of two different scales. With the incorporation of the worst recombination phase and the cuckoo search phase, the ITLBO algorithm can solve the multi-dimensional parameter optimization problems with a better search capability. Against blur, different illuminations, and occlusion conditions, the proposed fault detection method can still successfully identify 98.9% images of the field database, which demonstrates its reliability and superiority. However, there are still some images could not be identified properly. In addition, the procedures of Gabor transform, MLBP extraction and ITLBO search, can cause some computational burden. Future research should be focused on improving its performance, reducing its computation consumption, and applying it on other TFDS fault detection problems.

Acknowledgements

The authors are grateful to the Science and Technology Program of Harbin, China for financial support in this study.

Author Contributions

Hongjian Zhang, proposed the concept, performed the experiments and wrote the manuscript. Ping He, contributed to the generation of the primary ideas, designed the experiments and analyzed the data. Xudong Yang, participated in the data collection and algorithm simulation.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Zhu, Z.X.; Wang, G.Y. A fast potential fault regions locating method used in inspecting freight cars. *J. Comput.* **2014**, *9*, 1266–1273.
2. Zhou, K.; Lu, M.; Wang, G.; Ren, B.Y. Monitoring data transmission technology in 5T system based on message. *China Railw. Sci.* **2008**, *29*, 113–118.
3. Liu, Z.H.; Xiao, D.Y.; Chen, Y.M. Displacement fault detection of bearing weight saddle in TFDS based on Hough transform and symmetry validation. In Proceedings of the 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery, Chongqing, China, 29–31 May 2012; pp. 1404–1408.
4. Dai, P.; Yang, X.D.; He, P.; Zhang, H.J. Automatic recognition for losing of train bogie center plate screw based on multiple-fuzzy relation tree. In Proceedings of the 2009 International Workshop on Intelligent Systems and Applications, Wuhan, China, 23–24 May 2009; pp. 1–5.
5. Yang, X.D.; Ye, L.J.; Yuan, J.B. Research of computer vision fault recognition algorithm of center plate bolts of train. In Proceedings of the 2011 International Conference on Instrumentation, Measurement, Computer, Communication and Control, Beijing, China, 21–23 October 2011; pp. 978–981.
6. Li, N.; Wei, Z.Z.; Cao, Z.P.; Wei, X.G. Automatic fault recognition for losing of train bogie center plate bolt. In Proceedings of the 2012 IEEE 14th International Conference on Communication Technology, Chengdu, China, 9–11 November 2012; pp. 1001–1005.
7. Ahonen, T.; Hadid, A.; Pietikäinen, M. Face description with local binary patterns: Application to face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 2037–2041.
8. Guo, Z.H.; Zhang, L.; Zhang, D. A completed modeling of local binary pattern operator for texture classification. *IEEE Trans. Image Process.* **2010**, *19*, 1657–1663.
9. Benrachou, D.E.; Santos, F.N.; Boulebtateche, B.; Bensaoula, S. Automatic eye localization; multi-block LBP *vs.* Pyramidal LBP three-levels image decomposition for eye visual appearance description. *Lect. Notes Comput. Sci.* **2015**, *9117*, 718–726.
10. Pinjari, S.A.; Patil, N.N. A modified approach of fragile watermarking using Local Binary Pattern (LBP). In Proceedings of the 2015 International Conference on Pervasive Computing: Advance Communication Technology and Application for Society, Pune, India, 8–10 January 2015; pp. 1–4.
11. Monika; Kumar, M. A novel fingerprint minutiae matching using LBP. In Proceedings of the 2014 3rd International Conference on Reliability, Infocom Technologies and Optimization: Trends and Future Directions, Noida, India, 8–10 October 2014; pp. 1–4.

12. Bai, S. Sparse code LBP and SIFT features together for scene categorization. In Proceedings of the 2014 International Conference on Audio, Language and Image Processing, Shanghai, China, 7–9 July 2014; pp. 200–205.
13. Zhao, Q.Y.; Pan, B.C.; Pan, J.J.; Tang, Y.Y. Facial expression recognition based on fusion of Gabor and LBP features. In Proceedings of the 2008 International Conference on Wavelet Analysis and Pattern Recognition, Hong Kong, China, 30–31 August 2008; pp. 362–367.
14. Shen, L.; Bai, L.; Fairhurst, M. Gabor wavelets and general discriminant analysis for face identification and verification. *Image Vis. Comput.* **2007**, *25*, 553–563.
15. Wang, S.; Xia, Y.; Liu, Q.; Luo, J.; Zhu, Y.; Feng, D.D. Gabor feature based nonlocal means filter for textured image denoising. *J. Vis. Commun. Image Represent.* **2012**, *23*, 1008–1018.
16. Bissi, L.; Baruffa, G.; Placidi, P.; Ricci, E.; Scorzoni, A.; Valigi, P. Patch based yarn defect detection using Gabor filters. In Proceedings of the 9th IEEE International Instrumentation and Measurement Technology Conference, Graz, Austria, 13–16 May 2012; pp. 240–244.
17. Nouyed, I.; Amin, M.A.; Poon, B.; Yan, H. Human face recognition using weighted vote of Gabor magnitude filters. In Proceedings of the 7th International Conference on Information Technology and Application, Sydney, Australia, 21–24 November 2011; pp. 36–40.
18. Gao, T.; Ma, X.; Wang, J.G. Particle occlusion face recognition using adaptively weighted local Gabor filters. *J. Comput. Inf. Syst.* **2012**, *8*, 5271–5277.
19. Zhu, J.X.; Su, G.D.; Li, Y.C. Facial expression recognition based on Gabor feature and AdaBoost. *J. Optoelectron. Laser* **2006**, *17*, 993–998.
20. Rao, R.V.; Savsani, V.J.; Vakharia, D.P. Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems. *Comput. Aided Des.* **2011**, *43*, 303–315.
21. Rao, R.V.; Savsani, V.J.; Vakharia, D.P. Teaching-Learning-Based Optimization: An optimization method for continuous non-linear large scale problems. *Inf. Sci.* **2012**, *183*, 1–15.
22. Rao, R.V.; Patel, V. Multi-objective optimization of heat exchangers using a modified teaching-learning-based optimization algorithm. *Appl. Math. Model.* **2013**, *37*, 1147–1162.
23. Togan, V. Design of planar steel frames using teaching-learning based optimization. *Eng. Struct.* **2012**, *34*, 225–232.
24. Yu, K.J.; Wang, X.; Wang, Z.L. An improved teaching-learning-based optimization algorithm for numerical and engineering optimization problems. *J. Intell. Manuf.* **2014**, 1–13, doi:10.1007/s10845-014-0918-3.
25. Karaboga, D.; Akay, B. Artificial bee colony (ABC), harmony search and bees algorithms on numerical optimization. In Proceeding of IPROMS-2009 on Innovative Production Machines and Systems, Cardiff, UK, 24 March 2009; pp. 1–6.
26. Liu, H.; Cai, Z.X.; Wang, Y. Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. *Appl. Soft Comput.* **2010**, *10*, 629–640.
27. Mohamed, A.W.; Sabry, H.Z. Constrained optimization based on modified differential evolution algorithm. *Inf. Sci.* **2012**, *194*, 171–208.
28. Li, G.Q.; Niu, P.F.; Xiao, X.J. Development and investigation of efficient artificial bee colony algorithm for numerical function optimization. *Appl. Soft Comput.* **2012**, *12*, 320–332.
29. Brajevic, I.; Tuba, M. An upgrade artificial bee colony algorithm for constrained optimization problems. *J. Intell. Manuf.* **2013**, *24*, 729–740.

30. Rao, R.V.; Patel, V. An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems. *Int. J. Ind. Eng. Comput.* **2012**, *3*, 535–560.
31. Storn, R.; Price, K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359.
32. Yang, X.S.; Deb, S. Engineering optimization by Cuckoo Search. *Int. J. Math. Model. Numer. Optim.* **2010**, *4*, 330–343.
33. Ghasemi, M.; Ghavidel, S.; Gitizadeh, M.; Akbari, E.B. An improved teaching–Learning-based optimization algorithm using Lévy mutation strategy for non-smooth optimal power flow. *Int. J. Electr. Power Energy Syst.* **2015**, *65*, 375–384.
34. Elsayed, S.M.; Ruhul, A.S.; Daryl, L.E. A new genetic algorithm for solving optimization problems. *Eng. Appl. Artif. Intell.* **2014**, *27*, 57–69.
35. Tanweer, M.R.; Suresh, S.; Sundararajan, N. Self-regulating particle swarm optimization algorithm. *Inf. Sci.* **2015**, *294*, 182–202.
36. Črepinšek, M.; Liu, S.H.; Mernik, L. A note on teaching–Learning-based optimization algorithm. *Inf. Sci.* **2012**, *212*, 79–93.