

# Illumination correction via improved grey wolf optimizer for regularized random vector functional link network

Xiaochun Zhang<sup>1</sup>, and Zhiyu Zhou<sup>2\*</sup>

<sup>1</sup>Wenzhou Polytechnic, School of Intelligent Manufacturing  
Chashan Higher Education Park, Wenzhou, Zhejiang, China  
[e-mail: 2003030010@wzpt.edu.cn]

<sup>2</sup>Zhejiang Sci-Tech University, School of Computer Science and Technology  
Hangzhou, China  
[e-mail: zhouzhiyu1993@zstu.edu.cn]

\*Corresponding author: Zhiyu Zhou

*Received November 21, 2021; revised August 6, 2022; revised November 22, 2022; accepted February 20, 2023;  
published March 31, 2023*

---

## Abstract

In a random vector functional link (RVFL) network, shortcomings such as local optimal stagnation and decreased convergence performance cause a reduction in the accuracy of illumination correction by only inputting the weights and biases of hidden neurons. In this study, we proposed an improved regularized random vector functional link (RRVFL) network algorithm with an optimized grey wolf optimizer (GWO). Herein, we first proposed the moth-flame optimization (MFO) algorithm to provide a set of excellent initial populations to improve the convergence rate of GWO. Thereafter, the MFO-GWO algorithm simultaneously optimized the input feature, input weight, hidden node and bias of RRVFL, thereby avoiding local optimal stagnation. Finally, the MFO-GWO-RRVFL algorithm was applied to ameliorate the performance of illumination correction of various test images. The experimental results revealed that the MFO-GWO-RRVFL algorithm was stable, compatible, and exhibited a fast convergence rate.

---

**Keywords:** Grey Wolf Optimizer, Moth Flame Optimizer, Random Vector Functional Link Network, Illumination Correction

---

This work is supported by Zhejiang Provincial Department Education (No. FG2020056) and Zhejiang Provincial Key Research and Development Program (No. 2021C03013).

<http://doi.org/10.3837/tiis.2023.03.008>

ISSN : 1976-7277

## 1. Introduction

Illumination correction, also called illumination estimation or color constancy calculation [1-2], eliminates the influence of illumination on color, thereby obtaining the inherent color of the object itself. Although the conventional unsupervised illumination correction algorithm is uncomplicated and liable to achieve, it exhibits low performance when it does not meet the assumptions. Supervised illumination correction, a developed trend in the field of illumination correction, utilizes machine learning technology to construct the function of image color and illumination distribution, thereby providing better results.

Back propagation (BP) neural networks, extreme learning machine (ELM), and support vector regression (SVR) are commonly used in illumination correction research. A method of illumination correction via BP neural network proposed in [3] offers continuous output results; however, it is susceptible to local optimum stagnation. The image illumination correction algorithm based on SVR proposed in [4] can obtain high-precision learning accuracy for small samples; however, it involves cumbersome parameter adjustment. Li et al. [5] proposed an illumination correction algorithm with grey-edge and ELM. ELM can effectively overcome the deficiencies of the traditional neural networks; however, it is susceptible to parameter randomness. Zhou [6] applied regularization to solve the drawback of a pathological solution to the output weight of the traditional random vector functional connection (RVFL); however, it cannot resolve the implicit layer bias and the randomness of the input weight matrix.

Random weight networks such as ELM and RVFL exhibit fast learning. Although most traditional training techniques randomly select the connection weights and hidden biases for random weight networks, they still exhibit local optimization problems and degenerated convergence. In addition, adjusting the number of hidden neurons, determining the regularization factors, and selecting the appropriate transfer function in these kinds of random weight networks require time, energy, and human intervention.

The metaheuristic algorithm is a random search algorithm based on computational intelligence that solves the optimal solution of complex majorization issues; it is usually aroused by the evolution of nature and the ways of living organisms, such as an artificial bee colony (ABC) [7], particle swarm optimization (PSO) [8], grey wolf optimizer (GWO) [9], brain storm optimization (BSO) [10], and moth-flame optimization (MFO) [11]. Neural network optimization techniques based on the metaheuristic algorithm can avoid local optimality and exhibit good flexibility. Therefore, they offer additional room for improvement in the research of optimization algorithms, thereby resulting in the emergence of more competitive optimization algorithms when compared with the existing algorithms. Wang [12] proposed an improved whale algorithm to optimize the image color constancy calculation of SVR. Liu [13] proposed an RVFL illumination estimation algorithm that is optimized using the whale optimization algorithm. Zhou [14] used the good global search ability of the differential evolution (DE) algorithm to iteratively optimize the ELM and solve its issue of random parameter setting. Han [15] adopted the improved PSO algorithm and the More-Penrose generalized inverse algorithm to select parameters. Typically, in RVFL and ELM, optimization algorithms optimize only the input weight and biases of hidden neurons; they are unable to resolve local optimal stagnation and reduced convergence performance [16].

Inspired by the references [17-19], it is different from their optimization of only two parameters. This study proposes an illumination correction method using an improved grey

wolf optimizer to obtain a regularized random vector functional link (RRVFL) network. The master contributions of this paper are as follows:

- 1) The MFO optimizer provides a set of excellent initial populations for GWO to remove the effect of the population initialization and ameliorate the convergence speed and optimization effect of the GWO.
- 2) Since solely optimizing the parameters causes local optimal stagnation and reduced convergence performance, the MFO-GWO method is applied to simultaneously ameliorate the input characteristics, hidden nodes, bias, and input weights of the RRVFL network, thereby avoiding local optimal stagnation, obtaining reasonable accuracy results, and reducing the influence of human intervention to adjust the experimental parameters.
- 3) Analyzing the box diagram with the angle and chromaticity errors revealed that the proposed MFO-GWO-RRVFL algorithm exhibited good stability, nil outliers, and a small median error. In addition, it offers good predictive stability and excellent performance in the ten-fold cross-validation.
- 4) When compared with GMO-RRVFL, MFO-RRVFL, ABC-RRVFL, BSO-RRVFL, and PSO-RRVFL, the proposed MFO-GWO-RRVFL algorithm offers faster convergence speed and high illumination correction accuracy.

## 2. Related Work

In this section, we introduce principle of optimization algorithms and color constancy and narrate preparative job of color constancy.

### 2.1 Optimization algorithm

In engineering design, the optimization algorithm mainly selects a set of parameters (variables) to find the optimal value of the design index (target) among all the constraints. Before using the optimization algorithm to solve the actual problem, it is necessary to clarify the objective function (fitness function) required by the optimization algorithm, the parameters and constraints that need to be optimized. Optimization algorithms are mainly divided into exact methods (Exact Approaches) and heuristic algorithms.

The precise method [20] mainly proposes a mathematical model for a specific problem through mathematical modeling and solves the mathematical model through an optimization algorithm to get the optimal way of the issue. If the mathematical model is well designed, the solution obtained by the precise method can theoretically be optimal, but as the scale of the problem expands, this method becomes very laborious and cannot obtain the optimal solution in a limited time. S. Fateme Attar et al. [21] proposes a mixed integer programming model (MIP) to resolve the electric vehicle production routing issue.

A heuristic algorithm is a problem-oriented algorithm [22] that a feasible solution can be obtained under the constraints of space and time, but this is not necessarily the optimal solution. There are two types of heuristic algorithms, namely meta-heuristic algorithm, and traditional heuristic algorithm. Traditional heuristic algorithms include relaxation methods, stereotype methods, and steepest descent methods [23]. We improve on the traditional heuristic algorithm and get the meta-heuristic algorithm. This class of algorithms combines stochastic algorithms with local search. The difference between it and the traditional heuristic algorithm is whether there is a "random factor". A meta-heuristic [24] is an iterative process that continuously brings candidate solutions closer to the optimal solution with a strategy. The meta-heuristic algorithm needs to explore and develop the search stewardess,

explore to avoid the algorithm from entering the local optimal solution, and develop and improve the local search ability of the algorithm. Common meta-heuristic algorithms include ABC, PSO, GWO, MFO, and other optimization algorithms. These meta-heuristic optimization algorithms all contain the idea of swarm intelligence optimization and simulate the swarming behavior of animal populations in the real world. In fact, there has been a lot of work on improving the initial population of optimization algorithms, such as opposition based on learning [25], chaotic mapping [26] and Levi flight, etc. However, the initialization of these methods is not ideal, so we try to simply initialize the population with another optimization algorithm. Although this method will make the algorithm spend more time in the population initialization phase, this consumption is worth it, it can allow the subsequent optimization algorithm to obtain a better initial population, thereby effectively ameliorating the astringent speed of the entire arithmetic (the algorithm takes less time to reach convergence after initialization). Therefore, the computational cost of the entire algorithm does not increase significantly.

## 2.2 Color constancy

Color constancy [27] is an ability of the human visual adaptive system. Its main function is to ensure that the color perceived by vision maintains a relatively constant value under changing lighting conditions. The main methods to solve the problem of color constancy are partitioned into two Kinds: Statistics-based methods [28-32] and learning-based methods [12, 13, 33-36]. Statistical-based methods generally do not rely on the prior knowledge of the sample, and directly use the image information of the sample to estimate the illumination during image imaging. This method is fast, but the algorithm is not robust and suitable for ideal scenarios. The learning-based method makes full use of the training data to train the model. Although this method takes a lot of time to train the model, the model after the training has a relatively good performance. Wang [12] proposed a method based on a support vector machine to resolve the issue of color constancy. Partha et al. [35] applied the adversarial neural network to resolve the color constancy's issue, and proposed Color Constancy GANs (CC-GANs). There are three GANs methods used in the paper: Pix2Pix, CycleGAN, and StarGAN. et al. [36] applied multi-domain learning to solve the problem of color constancy. Multi-domain learning used training data from distinct equipment to train an uncomplicated model, learn complementary representations, and ameliorate generalization performance. They put forward a multi-domain color constancy method (MDICC).

## 3. Methodology

In this section, we discuss our raised illumination correction algorithm (MFO-GWO-RRVFL). We first introduce the theoretical underpinnings of the three components on which the method depends. Then, we'll describe how we made these three components work together to make up the overall illumination correction algorithm.

### 3.1 Regularized random vector functional link (RRVFL) network

RVFL has been developed from RVFL [37], it uses regularisation parameters and minimises training error to regulate the output weight.

$$\min_{\beta} C \|y - H\beta\|_2^2 + \|\beta\|_2^2 \quad (1)$$

After constraint optimization,

$$\min_{\beta} C \|e\|_2^2 + \|\beta\|_2^2 \text{ subject to } y - H\beta = e \quad (2)$$

Lagrange is defined as:

$$L(\beta, e, \lambda) = C \|e\|_2^2 + \|\beta\|_2^2 + \lambda^T (y - H\beta - e) \quad (3)$$

and contains  $N$  training error variables in  $e = [e_1, e_2, \dots, e_N]^T$ . According to the Karush-Kuhn-Tucker (KKT) theorem, we obtain

$$\beta = \left( H^T H C + \frac{I}{C} \right)^{-1} H^T y \quad (4)$$

If  $N < L$ ,

$$\beta = H^T \left( H^T H + \frac{I}{C} \right)^{-1} y \quad (5)$$

### 3.2 Grey wolf optimizer (GWO) and moth-flame optimization (MFO) algorithms

GWO has been aroused by the chase behaviour of grey wolves. The GWO algorithm identifies three leading wolves, denoted as  $\alpha$ ,  $\beta$ , and  $\delta$ , as the best solution, and directs the remaining wolves, denoted as  $\omega$ , to the region of alternative solutions, thereby finding the global solution. Wolf chase involves three primary steps: encircling, hunting, and assaulting the quarry. To simulate the event of the grey wolves surrounding prey, the positions of  $\alpha$ ,  $\beta$  or  $\delta$  are updated as follows:

$$D = |C \times X_p(t) - X(t)| \quad (6)$$

$$X(t+1) = X_p(t) - A \times D \quad (7)$$

where  $t$  represents the current iteration number,  $X_p(t)$  represents the current prey position,  $X(t)$  represents the current wolf position, and  $D$  represents the distance between the wolf and the prey. The coefficient vector  $A$  and  $C$  are calculated as follows:

$$A = 2ar_1 - a \quad (8)$$

$$C = 2r_2 \quad (9)$$

where  $r_1$  and  $r_2$  represent random vectors between 0 and 1, and  $a$  decreases linearly from 2 to 0 with an increase in iterations.

Hunting: By preserving the first three best solutions with  $\alpha$ ,  $\beta$ , and  $\delta$ , the grey wolf community updates its position as follows:

$$D_\alpha = |C_1 \cdot X_\alpha - X| \quad (10)$$

$$D_\beta = |C_2 \cdot X_\beta - X| \quad (11)$$

$$D_\delta = |C_3 \cdot X_\delta - X| \quad (12)$$

where  $X_\alpha$ ,  $X_\beta$ , and  $X_\delta$  represents the position of  $\alpha$ ,  $\beta$ , and  $\delta$ , respectively;  $X$  represents the location of the operating solution; and  $C_1$ ,  $C_2$ , and  $C_3$ , are randomly generated vectors. The aftermost location of the operating solution is shown as follows:

$$X_1 = X_\alpha - A_1 \cdot D_\alpha \quad (13)$$

$$X_2 = X_\beta - A_2 \cdot D_\beta \quad (14)$$

$$X_3 = X_\delta - A_3 \cdot D_\delta \quad (15)$$

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3} \quad (16)$$

where,  $A_1$ ,  $A_2$ , and  $A_3$  are random vectors, and  $t$  represents the number of iterations. MFO utilizes two populations of moth and flame to respectively represent the candidate solution and the optimal solution of the algorithm. To avoid letting GWO fall into a local optimum, this study proposes the utilization of the MFO algorithm to initialise the GWO algorithm population.

### 3.3 Proposed illumination correction algorithm

The proposed MFO-GWO-RRVFL algorithm uses MFO-GWO to simultaneously optimize the input feature and weight, hidden bias and node of RRVFL. The MFO in MFO-GWO provides an excellent initial population for GWO, thereby improving its convergence speed. Primarily, the optimization process determines the number of parameters that can be optimized by pre-setting the maximum network of RRVFL. Thereafter, the corresponding hidden bias and input weights are selected via effective hidden nodes and input nodes, respectively. The proposed algorithm was obtained by combining MFO-GWO and the parameter optimization method to subsequently optimize RRVFL.

In RRVFL, the random generation of parameters leads to a poor training effect. To optimise this shortcoming, an optimisation algorithm can conduct an automatic search for optimal parameters and ameliorate the prediction performance of the RRVFL network. In this study, the root mean square error (RMSE) of the predicted value and the true value was selected as the fitness function and expressed as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^N \|(\sum_{j=1}^L \beta_j g(w_j X_i + b_j) + \sum_{j=L+1}^{L+d} \beta_j X_{ij}) - t_j\|_2^2}{m \times N}} \quad (17)$$

where  $d$ ,  $L$ ,  $m$ , and  $N$  represent the quantity of RRVFL hidden layer nodes, input nodes, output vectors, and samples, respectively.

The flow chart of the MFO-GWO-RRVFL algorithm is displayed in [Fig. 1](#). In the algorithm process of MFO-GWO-RRVFL, the parameters of the optimised algorithm population corresponding to RRVFL were primarily completed through mapping and screening; [Fig. 2](#) is a schematic diagram of its structure. The input and hidden nodes were mapped as 0 or 1 by rounding the number between  $[-1, 1]$ . If the obtained result was 1, the feature or node was used (valid node); if the obtained result was 0, the feature or node was not used (invalid node). The input weights and implicit layer bias matrices were screened and rearranged according to the valid features or nodes (the all-black matrix displayed on the right side in [Fig. 2](#)). The optimisation goal was completed after reaching the maximum number of iterations. Thereafter, the abovementioned mapping and screening methods were utilized to select the effective parameters and obtain the optimised illumination prediction model.

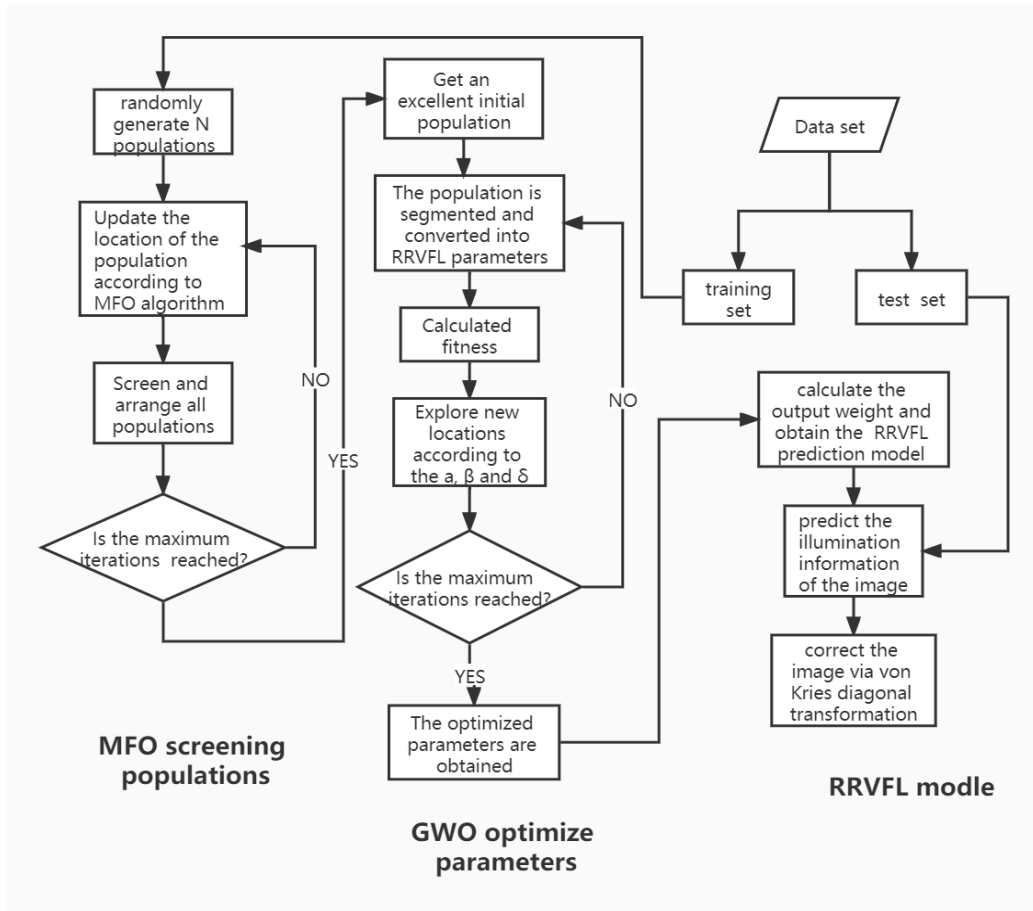


Fig. 1. Flow chart of the proposed MFO-GWO-RRVFL algorithm.

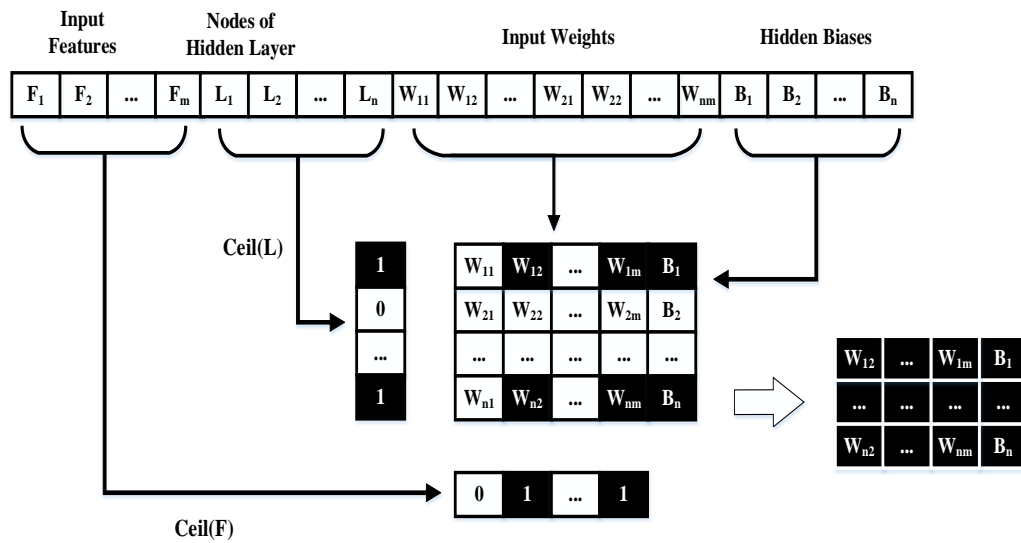


Fig. 2. Population structure and parameters represented by each segment of the MFO-GWO-RRVFL algorithm.

## 4. Experimental results and analysis

We selected the GWO-RRVFL, MFO-RRVFL, ABC-RRVFL, BSO-RRVFL, PSO-RRVFL, RRVFL, RELM, BP, and SVR algorithms for comparison with proposed MFO-GWO-RRVFL illumination correction algorithm. The process for optimising RRVFL using GWO, MFO, ABC, BSO, and PSO was identical to MFO-GWO.

### 4.1 Pre-processing of data sets

From the SFU Lab [38] datasets, 321 images were taken from 31 laboratory scenes under 11 different light sources, including three different fluorescent lamps, four incandescent lamps, and four incandescent lamps with colour filters. We employed the grey edge algorithm as an efficient feature extraction framework rather than a calculation method. By changing the parameters, the chromaticity value was calculated as a feature based on the current parameters. In this experiment, for  $n \in \{0,1,2\}$ ,  $p \in \{1,2,\dots,10\}$ , and  $\sigma = \{1,3,5,7,9\}$ , 150 values could be obtained for each chromaticity. Therefore, a total of 300 features could be obtained for  $r$  and  $g$  chromaticity.

### 4.2 Performance evaluation indicators

In the experiment, 10 algorithms were trained to obtain the corresponding prediction models, and each model was estimated the chromaticity of illumination. Then, the chromaticity error and angle error of each algorithm are calculated to assess the property of every algorithm. Finally, the angle error and chromaticity error were statistically analysed, and four statistical values were obtained: the error's mean (Mean), the error's median (Median), the average of the optimal 25% (Best25), and the average of the worst 25% (Worst25).

### 4.3 Experimental parameter setting

#### 4.3.1 Parameter selection of MFO-GWO-RRVFL algorithm

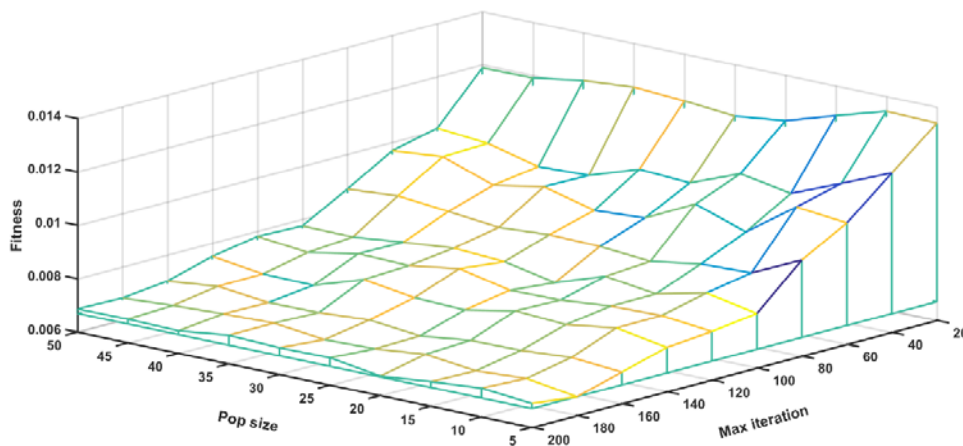
To assess the property of the raised MFO-GWO-RRVFL model, we first determined the parameter optimisation range of the RRVFL network. The MFO-GWO algorithm optimised four parameters for RRVFL, namely input feature, hidden node and bias, and input weight. The maximum dimension of the hidden bias and input weight was determined by the upper search limit of the input feature and the hidden node, respectively. Using the grey edge algorithm, we abstracted a whole of 300 dimensional traits; therefore, the maximum search of the input node was 300. The efficient feature training model in the 300-dimensional feature was automatically selected by the optimisation algorithm. The maximum hidden nodes was 600. The activation function and the regularisation coefficients were set to sigmoid and 100, respectively. We determined the population number and the largest quantity of iterations as they both affect the optimisation effect of the MFO-GWO algorithm. Conventionally, a larger population and a larger number of maximum iterations correspond to a better convergence effect. In addition, both parameters are proportional to the calculation time. Therefore, in the practical application process, to obtain a good optimisation effect, a minimum number of population and maximum iterations should be selected. We selected the value range of the population number and the maximum iteration to be  $\{5, 10, \dots, 50\}$  and  $\{20, 40, \dots, 200\}$ , respectively. We performed three experiments for each parameter combination to obtain the average fitness values, as displayed in [Table 1](#).



**Table 1.** Average fitness under different population numbers and iterations.

Max iteration										
Pop size	20	40	60	80	100	120	140	160	180	200
5	0.01339	0.01197	0.01054	0.00960	0.00804	0.00781	0.00766	0.00718	0.00670	0.00689
10	0.01344	0.01118	0.01074	0.00872	0.00841	0.00801	0.00799	0.00733	0.00699	0.00707
15	0.01288	0.01041	0.00941	0.00866	0.00798	0.00797	0.00772	0.00720	0.00688	0.00692
20	0.01229	0.01075	0.01006	0.00836	0.00824	0.00780	0.00784	0.00718	0.00710	0.00673
25	0.01207	0.01001	0.00914	0.00852	0.00761	0.00777	0.00748	0.00695	0.00691	0.00702
30	0.01224	0.01012	0.00903	0.00856	0.00800	0.00818	0.00754	0.00715	0.00686	0.00700
35	0.01236	0.00950	0.00952	0.00844	0.00811	0.00790	0.00782	0.00703	0.00708	0.00707
40	0.01221	0.00943	0.00918	0.00864	0.00793	0.00795	0.00721	0.00701	0.00703	0.00684
45	0.01191	0.00990	0.00985	0.00883	0.00773	0.00760	0.00724	0.00722	0.00695	0.00688
50	0.01189	0.01008	0.00968	0.00869	0.00775	0.00780	0.00750	0.00702	0.00683	0.00687

**Fig. 3** illustrates the changes in the fitness of the proposed MFO-GWO-RRVFL algorithm with distinct parameter combinations. As per the figure, the value of the population number had negligible influence on the algorithm fitness. Although when the population number increased from 5 to 20, the fitness decreased significantly, when the population number was greater than 20, it barely affected the fitness. Therefore, to obtain a better optimisation effect, the population number was taken as 20. In contrast, the value of maximum iterations had a significant impact on the fitness. When the maximum quantity of iterations increased from 20 to 200, the fitness decreased continuously and eventually decreased by approximately 50%.

**Fig. 3.** Changes in algorithm fitness under distinct values of population numbers and maximum iterations.

To get the suitable quantity of maximum iterations, we constructed a fitness contour map as displayed in **Fig. 4**. As per the figure, when the value of maximum iterations increased from 20 to 100, the fitness contour lines were denser, indicating a rapidly decreasing fitness. When this value increased from 100 to 200, the fitness contour lines were

sparse, suggesting that the fitness decreased slowly in this process. In addition, when the values of maximum iterations were 100 and 200, the running time of the algorithm doubled. Therefore, considering the optimisation effect and algorithm running time, the value for maximum number of iterations was set to 100.

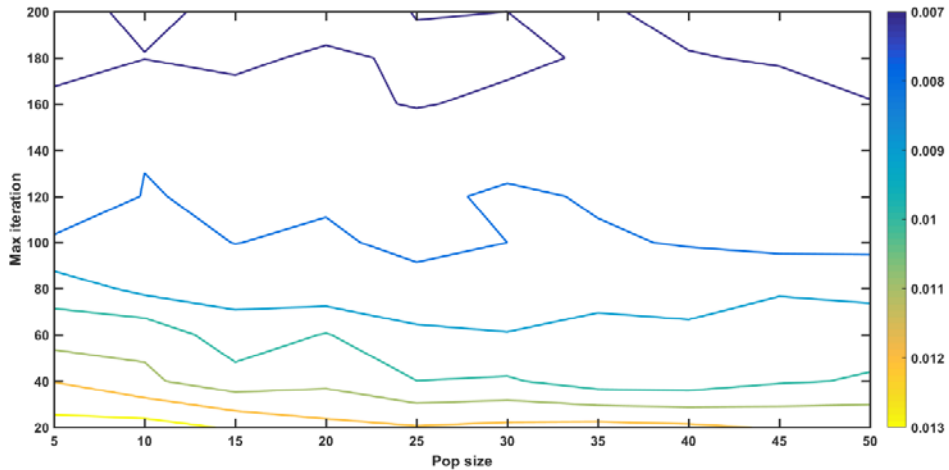


Fig. 4. Fitness contour lines under different values of population numbers and maximum iterations.

### 4.3.2 The other algorithms' parameter selection

To make sure fairness in the experimental outcomes, we attempted to ensure the consistency of the parameters for the comparison algorithms. Like MFO-GWO, for GWO-RRVFL, MFO-RRVFL, ABC-RRVFL, BSO-RRVFL, and PSO-RRVFL, the optimum value of population number and maximum iteration of GWO, MFO, ABC, BSO, and PSO are set to 20 and 100, respectively. The search upper limit of the hidden layer node and the input feature were consistent with the MFO-GWO-RRVFL algorithm as well, at 300 and 600, respectively. The parameter settings of RRVFL, RELM, BP, and SVR algorithms have been summarised in Table 2. For the other parameters, the default values of the MATLAB toolbox were employed.

Table 2. Some parameters of RRVFL, RELM, BP and SVR.

Algorithm	Parameter	Value
RRVFL	Number of hidden layer nodes	600
	Activation function	Sigmoid
	Regularisation coefficient	100
RELM	Number of hidden layer nodes	600
	Activation function	Sigmoid
	Regularisation coefficient	100
BP	Learning rate	0.05
	Training times	600
SVR	Kernel function	RBF
	Penalty parameter	2.2

## 4.4 Discussion of experimental results

### 4.4.1 Comparison of the illumination correction effect

To confirm the illumination correction effect of the proposed MFO-GWO-RRVFL algorithm, we first trained these predictive models on the training set of three datasets and then used the trained predictive model to predict the test set of the three datasets, and finally recorded the prediction results and calculated the angular error, chromaticity error, and the four statistical values of the error of each illumination prediction model on the SFU Lab datasets.

As can be seen from **Tables 3-4**, the angle and chromaticity errors of the proposed MFO-GWO-RRVFL algorithm were the lowest in most indexes, except for the median chromaticity error being slightly greater than that of the ABC-RRVFL, GWO-RRVFL, and PSO-RRVFL algorithms. In terms of angle error, the mean errors of the proposed MFO-GWO-RRVFL algorithm were 21.5%, 9.8%, 28.3%, 36.1%, 23.9%, 44.3%, 45.3%, 65.1%, and 58.8% lower than those of the GMO-RRVFL, MFO-RRVFL, ABC-RRVFL, BSO-RRVFL, PSO-RRVFL, RRVFL, RELM, BP and SVR methods, respectively. On chromaticity error, the average error of the proposed MFO-GWO-RRVFL algorithm was lower than the GMO-RRVFL, MFO-RRVFL, ABC-RRVFL, BSO-RRVFL, PSO-RRVFL, RRVFL, RELM, BP and SVR algorithms by 21.1%, 11.4%, 31.1%, 34.9%, 26.6%, 44.2%, 43.2%, 63.5%, and 56.4%, respectively.

**Table 3.** Angle errors comparisons on SFU lab (minimum values are displayed in bold)

Algorithm	Mean	Median	Best25	Worst25
MFO-GWO-RRVFL	<b>1.190</b>	1.051	<b>1.518</b>	<b>2.459</b>
GWO-RRVFL	1.516	1.068	1.980	3.600
MFO-RRVFL	1.319	1.074	1.659	2.673
ABC-RRVFL	1.660	<b>0.884</b>	2.133	4.303
BSO-RRVFL	1.862	1.074	2.422	4.581
PSO-RRVFL	1.564	0.992	2.008	3.593
RRVFL	2.137	1.091	2.764	5.697
RELM	2.175	1.151	2.855	5.924
BP	3.406	2.779	4.194	6.671
SVR	2.890	2.261	3.656	6.355

**Table 4.** Chromaticity error comparisons on SFU lab (minimum value is displayed in bold)

Algorithm	Mean	Median	Best25	Worst25
MFO-GWO-RRVFL	<b>9.484E-03</b>	8.025E-03	<b>1.192E-02</b>	<b>1.895E-02</b>
GWO-RRVFL	1.202E-02	7.072E-03	1.563E-02	2.924E-02
MFO-RRVFL	1.070E-02	9.322E-03	1.346E-02	2.111E-02
ABC-RRVFL	1.376E-02	<b>6.119E-03</b>	1.768E-02	3.618E-02
BSO-RRVFL	1.456E-02	8.418E-03	1.891E-02	3.673E-02
PSO-RRVFL	1.292E-02	7.350E-03	1.668E-02	3.091E-02
RRVFL	1.700E-02	9.053E-03	2.184E-02	4.440E-02
RELM	1.671E-02	9.591E-03	2.182E-02	4.411E-02
BP	2.602E-02	2.078E-02	3.178E-02	5.155E-02
SVR	2.175E-02	1.624E-02	2.733E-02	4.567E-02

**Figs. 5-14** display the line graph of the illumination chromaticity prediction results the actual results of RRVFL optimized with MFO-GWO, GWO, MFO, ABC, BSO, PSO respectively, RRVFL, RELM, BP, and SVR algorithms on the SFU Lab test set, respectively. For all the figures, (a) and (b) display the predicted results of the  $r$  and  $g$  chromaticity, respectively. According to the figures, the prediction accuracy of the proposed MFO-GWO-

RRVFL algorithm was higher than the other algorithms. In addition, the predicted values of the  $r$  chromaticity and  $g$  chromaticity deviated negligibly from the real values. To evaluate the prediction errors of each algorithm on the test set more intuitively, the bands of  $r$  chromaticity and  $g$  chromaticity prediction errors were drawn for all the algorithms, as displayed in Fig. 15 and Fig. 16, respectively. As expected, the proposed MFO-GWO-RRVFL algorithm exhibited excellent prediction performance. The  $r$  and  $g$  chromaticity prediction errors for all test samples were quite small. When compared with the other algorithms, the error fluctuation was gentle, thereby indicating a stable prediction effect. It was rare for the predicted value to deviate extensively from the real value in a certain test sample, like other algorithms.

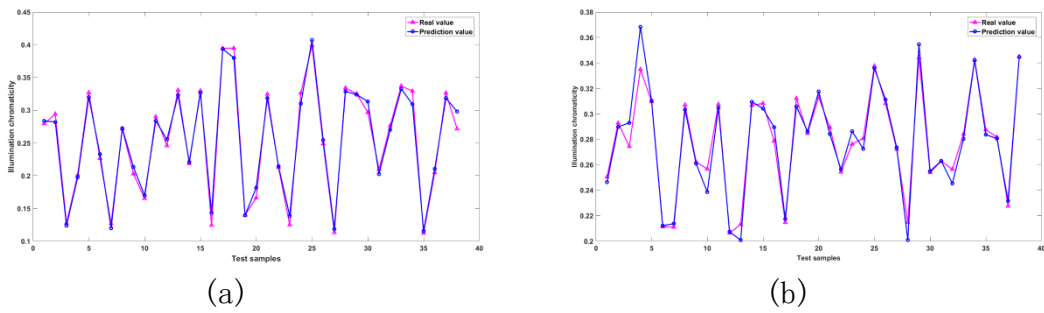


Fig. 5. Illumination prediction results of MFO-GWO-RRVFL.

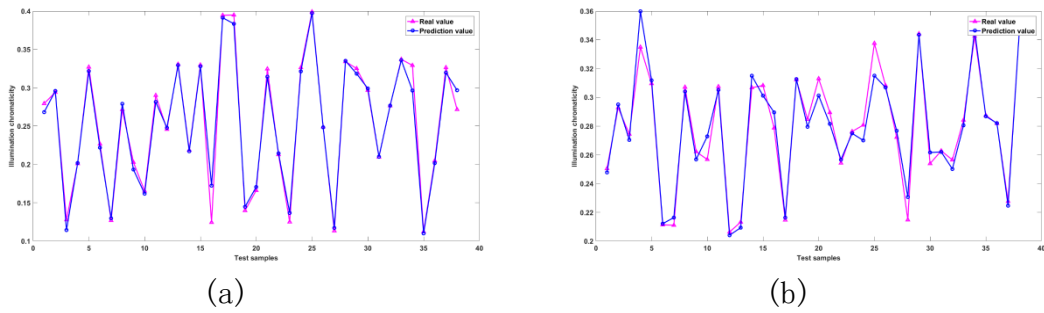


Fig. 6. Illumination prediction results of GWO-RRVFL.

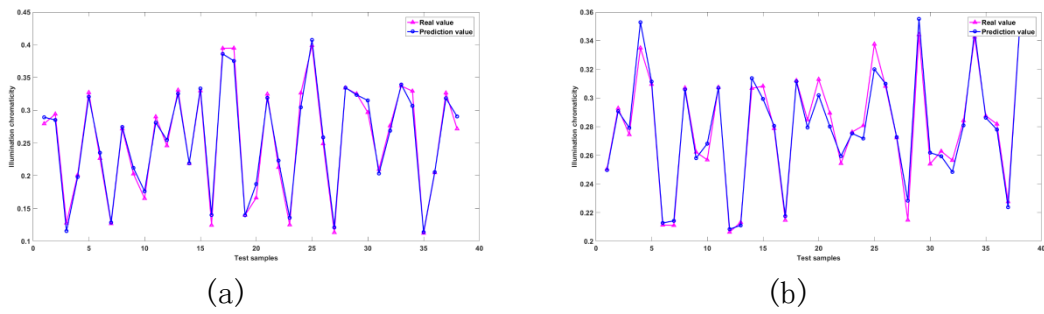
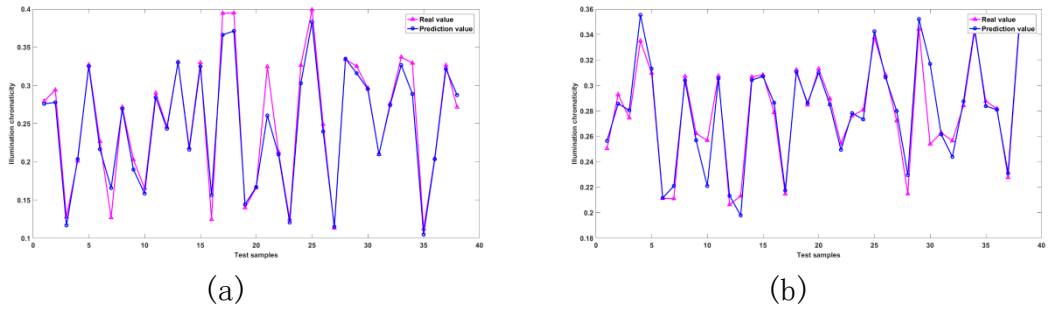
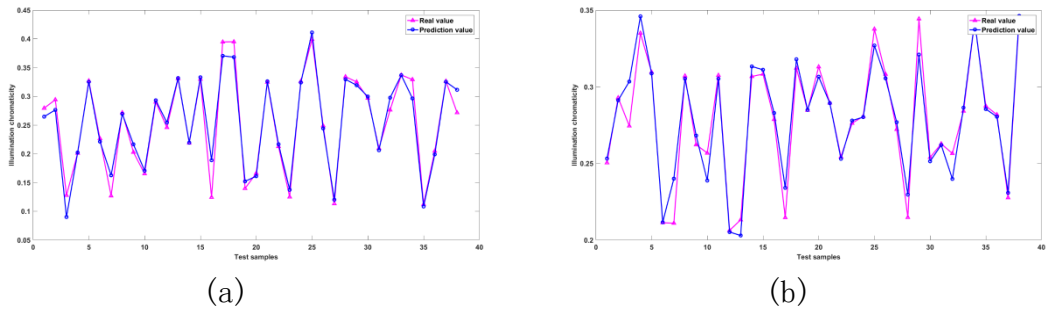


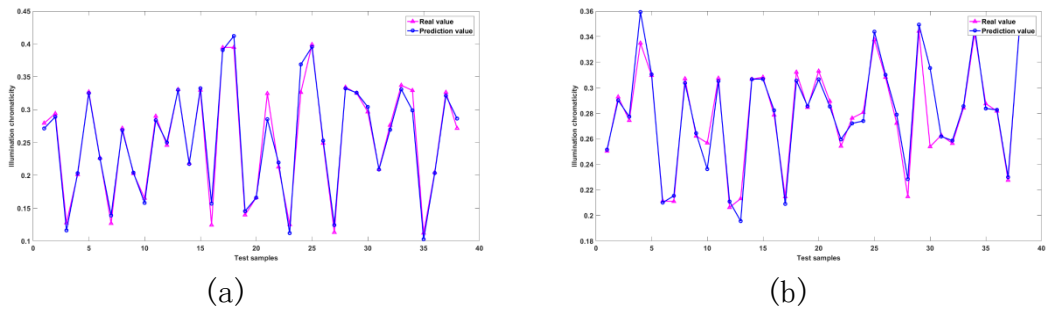
Fig. 7. Illumination prediction results of MFO-RRVFL.



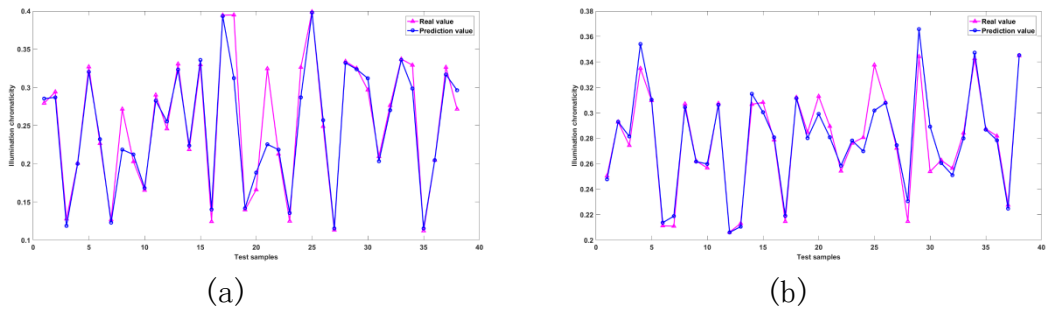
**Fig. 8.** Illumination prediction results of ABC-RRVFL.



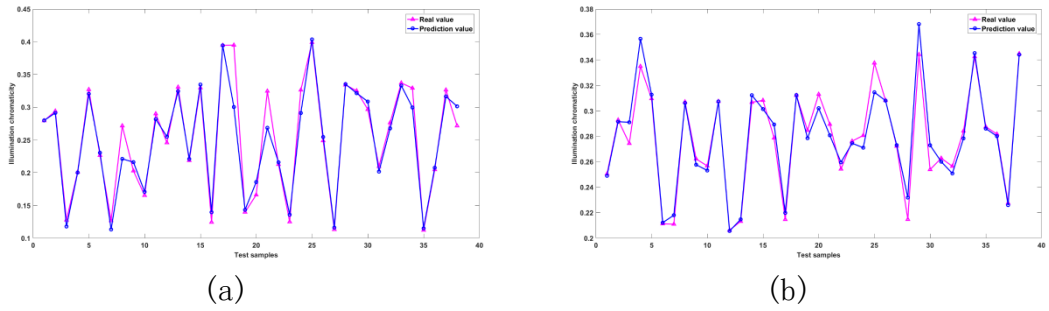
**Fig. 9.** Illumination prediction results of BSO-RRVFL.



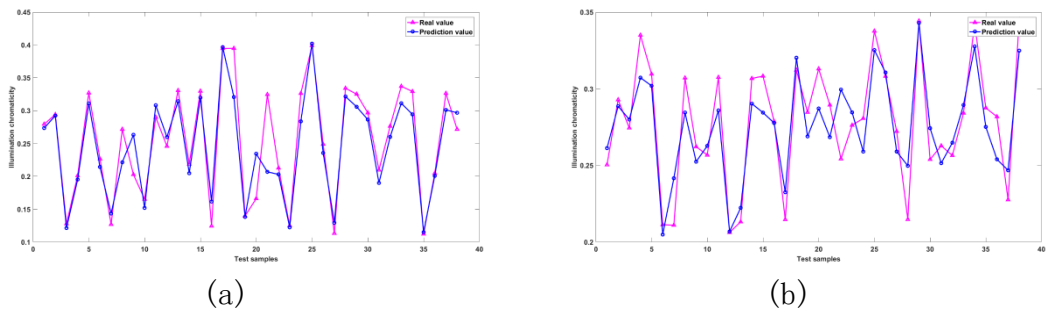
**Fig. 10.** Illumination prediction results of PSO-RRVFL.



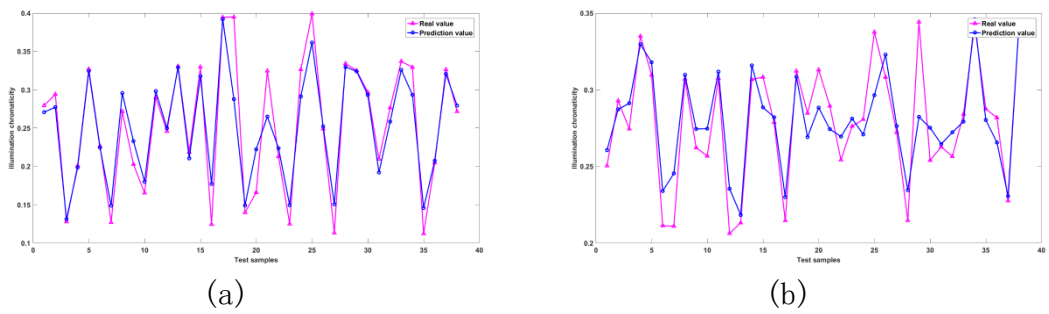
**Fig. 11.** Illumination prediction results of RRVFL.



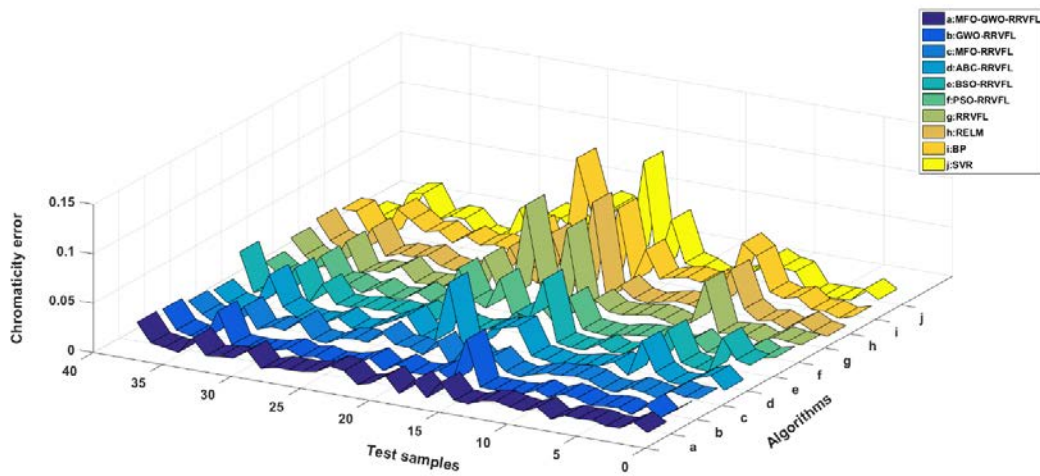
**Fig. 12.** Illumination prediction results of RELM.



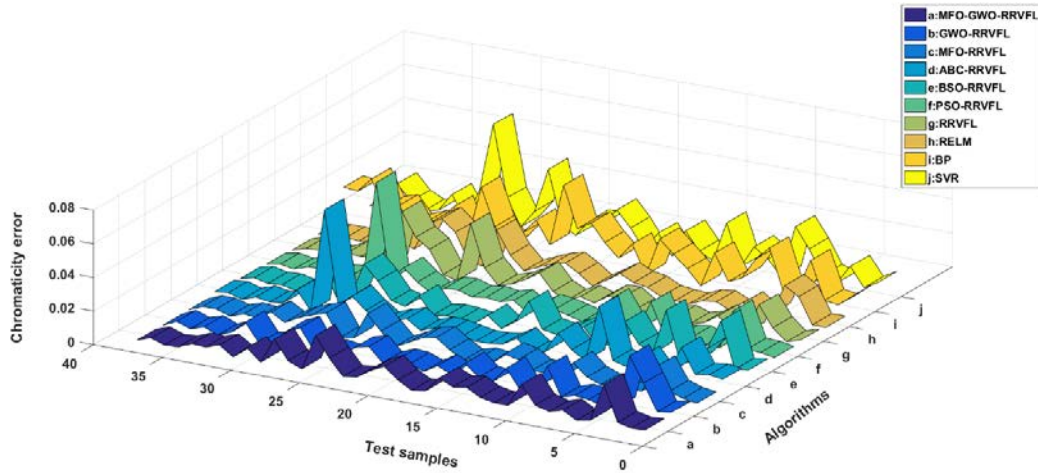
**Fig. 13.** Illumination prediction results of BP.



**Fig. 14.** Illumination prediction results of SVR.



**Fig. 15.** Bands of  $r$  chromaticity error of all individual algorithms.

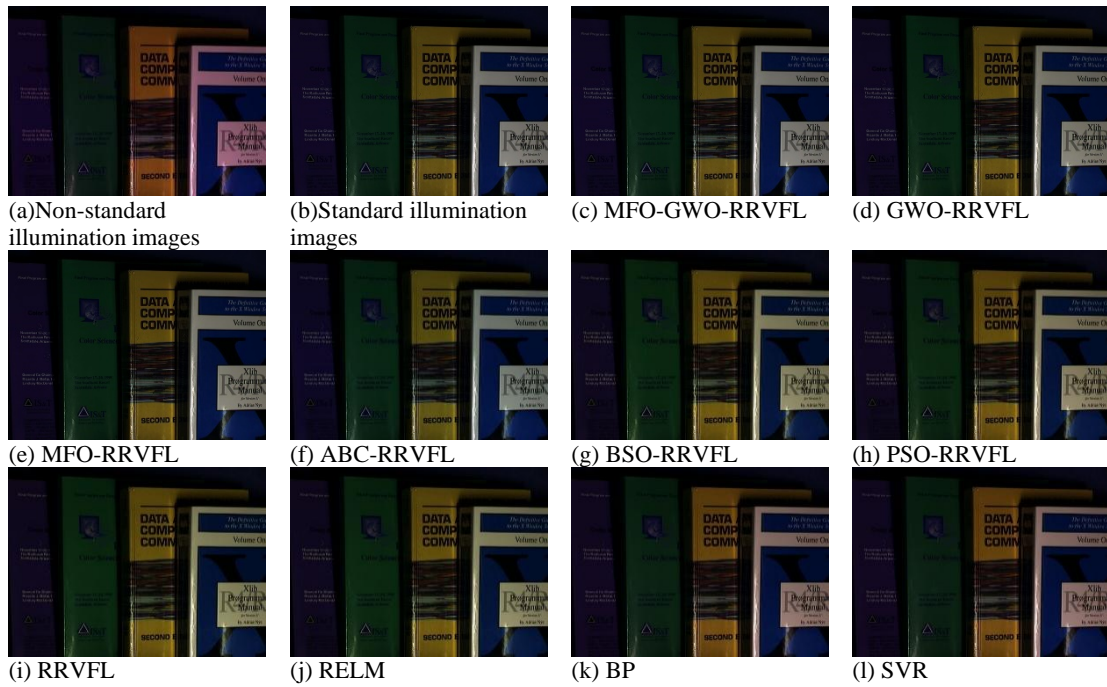


**Fig. 16.** Bands of  $g$  chromaticity error of all individual algorithms.

In conclusion, the proposed MFO-GWO-RRVFL algorithm exhibited favourable prediction performance. When compared with the MFO-RRVFL and GWO-RRVFL algorithms, the prediction performance had improved to a lesser extent; however, when compared with the ABC-RRVFL, BSO-RRVFL, PSO-RRVFL, RELM, RRVFL, BP, and SVR, the prediction performance had improved substantially.

#### 4.4.2 Comparison of image correction effects

After predicting the chromaticity of the test set samples according to the trained model, we used the von Kries diagonal transformation to correct the image under non-standard illumination to an image under standard illumination. To display the correction effect directly, this study took a test set image as an example. **Fig. 17** shows the non-standard illumination image, standard illumination image, and the illumination corrected image using the MFO-GWO-RRVFL, MFO-RRVFL, GWO-RRVFL, ABC-RRVFL, BSO-RRVFL, PSO-RRVFL, RRVFL, RELM, BP, and SVR, respectively. In terms of the test set picture, the proposed MFO-GWO-RRVFL algorithm, and the MFO-RRVFL and GWO-RRVFL algorithms portrayed a better correction effect, essentially eliminating the influence of illumination colour. However, images after illumination correction using ABC-RRVFL, BSO-RRVFL, PSO-RRVFL, RRVFL, and RELM appeared green, whereas the images after illumination correction using BP and SVR algorithms appeared red. The reddened effect was considered more unfavourable.



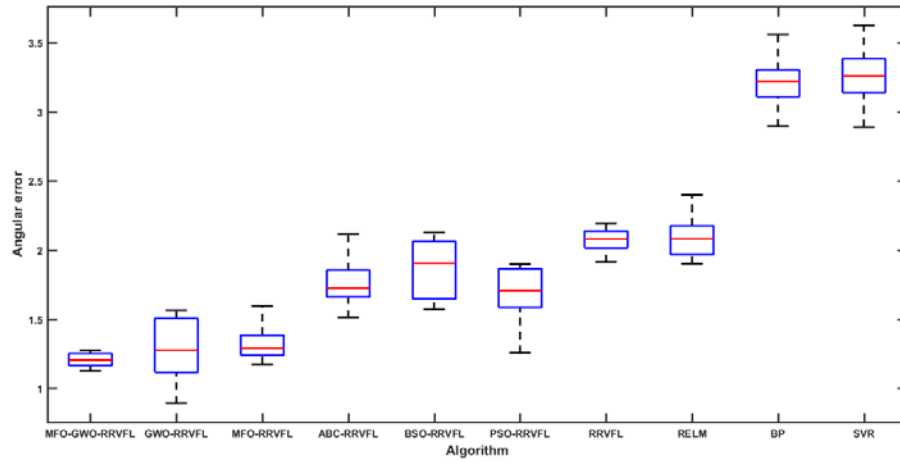
**Fig. 17.** Effect of each algorithm on image correction.

## 4.5 Stability and convergence of the algorithm

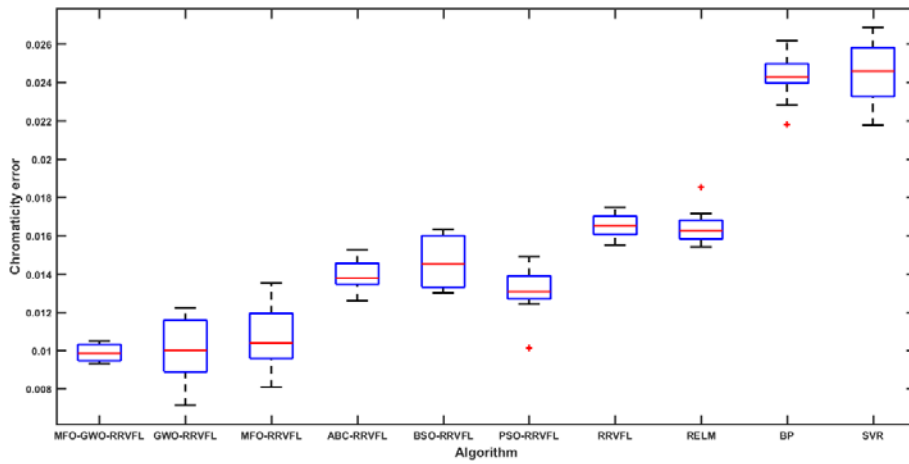
### 4.5.1 Analysis of stability

To analyse proposed MFO-GWO-RRVFL algorithm and other comparison algorithms, we employed a tenfold cross-validation means to test all the algorithms. The dataset was grouped into 10 folds, and in turn, nine folds were used for training. The remaining one-fold was utilized for predicting and calculating the average angle blunder and the chromaticity blunder. The respective angle and chromaticity error box diagram drawn according to the experimental results are displayed in [Fig. 18](#) and [Fig. 19](#). The blue illustrations in each figure represent the upper and lower quartiles, and encased within them is the corresponding error distribution. The height of the blue box is inversely proportional to the density of the data error distribution. As a result, the angle and chromaticity errors distribution of the MFO-GWO-RRVFL, ABC-RRVFL, and PSO-RRVFL algorithms were more concentrated, thereby exhibiting good stability. Next, the position of the red markers (outliers) and red lines (median errors) were analysed. The proposed MFO-GWO-RRVFL algorithm exhibited good predictive stability with no outliers. In addition, its median error was lower than the other comparison algorithms. Overall, the proposed algorithm exhibited the best overall performance in the ten-fold cross-validation.





**Fig. 18.** Angle error box diagram of each algorithm.



**Fig. 19.** Chromaticity error box diagram of each algorithm.

**Table 5** is obtained by calculating the standard deviation. From **Table 9**, we can observe that the standard deviation of the two error indicators of our proposed method is the smallest, and there is a certain gap in the second place, which a difference is 0.046 and 1.938E-04 respectively. This again proves that our proposed method has good robustness.

**Table 5.** STDEV of angle error and chromaticity error

Algorithm	STDEV of angle error	STDEV of chromaticity error
MFO-GWO-RRVFL	<b>0.050</b>	<b>4.461E-04</b>
GWO-RRVFL	0.214	1.618E-03
MFO-RRVFL	0.129	1.718E-03
ABC-RRVFL	0.172	8.917E-04
BSO-RRVFL	0.216	1.299E-03
PSO-RRVFL	0.197	1.247E-03
RRVFL	0.096	6.399E-04
RELM	0.146	9.231E-04
BP	0.180	1.256E-03
SVR	0.217	1.695E-03

### 4.5.2 Iterative curve analysis for convergence

The convergence speed and effect are important criteria for evaluating an optimisation algorithm. Therefore, to analyse the convergence performance of the proposed algorithm, we set the iterations' maximum quantity to 200, and each of the MFO-GWO-RRVFL, MFO-RRVFL, GWO-RRVFL, ABC-RRVFL, BSO-RRVFL, and PSO-RRVFL were run 10 times. The final convergence results were sorted from large to small, and the experimental results in the fifth place were selected to draw the convergence curve. Fig. 20 displays the changes in the fitness of each algorithm during the optimisation process. As per the figure, the proposed MFO-GWO-RRVFL algorithm exhibited the fastest convergence speed. Although low fitness was achieved after 20 iterations, the convergence effect was the best after 200 iterations, thereby suggesting that the algorithm could always find better parameters when compared with other algorithms during optimisation; this is beneficial for training a better illumination prediction model.

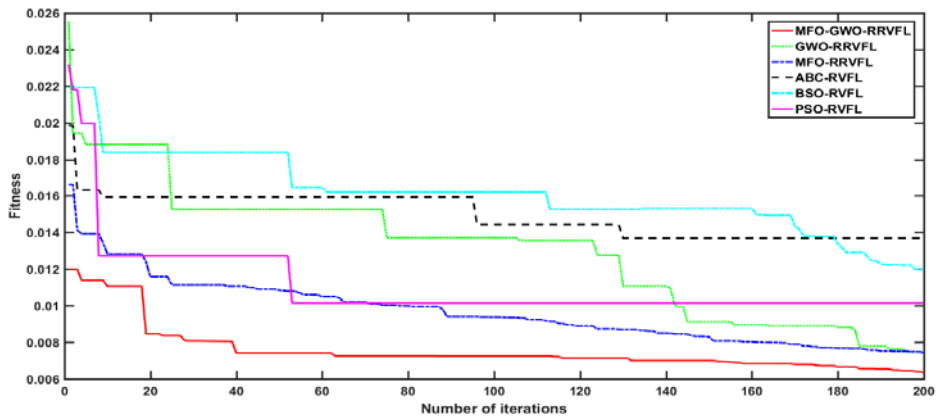


Fig. 20. Comparison of the convergence speed of optimisation algorithms.

### 4.6 Comparison of the significance level parameter of the algorithms

Table 6. Wilcoxon signed rank test results of angle errors of each algorithm ( $p$  values less than 0.05 are underlined).

Proposed algorithm	Compared algorithm	$p$ value
MFO-GWO-RRVFL	GWO-RRVFL	0.175
	MFO-RRVFL	<u>3.877E-02</u>
	ABC-RRVFL	<u>1.1865E-02</u>
	BSO-RRVFL	<u>2.161E-03</u>
	PSO-RRVFL	<u>1.514E-02</u>
	RRVFL	<u>1.186E-02</u>
	RELM	<u>3.993E-03</u>
	BP	<u>2.000E-07</u>
SVR	<u>1.882E-04</u>	

To analyse whether the prediction performance of the proposed MFO-GWO-RRVFL was substantially different than the other algorithms, we employed the Wilcoxon signed rank test to calculate the angular errors of all the algorithms in the test samples. The parameter of significance level  $p$  was set to 0.05. We use the Wilcoxon signed-rank test to compare the results of the two algorithms,  $p$  was obtained as less than 0.05, thereby indicating that the two algorithms' error distributions were substantially different. Table 5 presents the test

results for the angle errors of each algorithm for the same test set. As per the table, the prediction error distribution of the proposed MFO-GWO-RRVFL algorithm was considerably different from that of other algorithms except GWO-RRVFL, whereas its prediction performance of compared to GWO-RRVFL on the test set was somewhat similar.

## 5. Discussion

In order to verify the proposed method in different scenarios, we conducted related experiments on three other datasets: Simple Cube++ [39], Colour Checker [40] and SFU Grey-ball [41].

**Simple Cube++:** The Simple Cube++ contains 2234 images from Cube++. It is a small and simple version of Cube++, but it contains all the scenarios that cube++ has, and all the pictures were downscaled to 648\*432 size, and the dataset changed from the original 200G to 2G.

**Colour Checker:** The Colour Checker dataset contains 568 high-quality images captured by two models of Canon cameras in two natural scenes, indoor and outdoor. A standard 24-color chart is placed in each of these images, and information about the position of the color card in the image is provided.

**SFU Grey-ball:** The SFU Grey-ball dataset is a large image set contains more than 11000 images of indoor and outdoor lighting scenes. The light source color value is obtained through a gray ball bound to the camera. The source data is about 2 hours' video of outdoor and indoor daily scenes in Vancouver and Scottsdale captured by Sony VX-2000 digital camera.

We divide the three datasets into training sets and training sets, where the original Simple Cube++ has been divided into original datasets and the Colour Checker and the SFU Grey-ball divide the datasets 9:1. Then we performed the same experiment as 4.3.1 and got the test results as shown in **Tables 7-12**.

**Simple Cube++:**

As can be seen from **Tables 7-9**, the SVR algorithm has obtained the best results in all indicators and seems to perform perfectly better in Simple Cube++ than the algorithm proposed. The algorithm we proposed won second place in all indicators. Among them, the average angle error and the average Chromaticity error differ from the SVR algorithm by 0.419 and 3.47E-03, respectively.

**Table 7.** Angle errors comparisons on Simple Cube++

Algorithm	Mean	Median	Best25	Worst25
MFO-GWO-RRVFL	2.681	1.854	0.498	6.124
GWO-RRVFL	3.097	2.187	0.717	6.882
MFO-RRVFL	2.699	1.914	0.495	6.136
ABC-RRVFL	3.473	2.397	0.762	8.025
BSO-RRVFL	2.770	2.013	0.616	6.286
PSO-RRVFL	3.339	2.301	0.645	7.795
RRVFL	2.765	1.930	0.537	6.299
RELM	2.754	1.925	0.508	6.295
BP	2.725	1.873	0.564	6.236
SVR	<b>2.262</b>	<b>1.551</b>	<b>0.465</b>	<b>5.451</b>

**Table 8.** Chromaticity errors comparisons on Simple Cube++

Algorithm	Mean	Median	Best25	Worst25
MFO-GWO-RRVFL	1.931E-02	1.389E-02	3.810E-03	4.348E-02
GWO-RRVFL	2.266E-02	1.649E-02	5.703E-03	4.913E-02
MFO-RRVFL	1.945E-02	1.428E-02	3.800E-03	4.346E-02
ABC-RRVFL	2.579E-02	1.795E-02	5.865E-03	5.948E-02
BSO-RRVFL	2.002E-02	1.439E-02	4.726E-03	4.485E-02
PSO-RRVFL	2.451E-02	1.748E-02	5.115E-03	5.694E-02
RRVFL	2.006E-02	1.436E-02	4.139E-03	4.501E-02
RELM	1.994E-02	1.430E-02	3.811E-03	4.493E-02
BP	1.985E-02	1.499E-02	4.679E-03	4.349E-02
SVR	<b>1.584E-02</b>	<b>1.073E-02</b>	<b>3.656E-03</b>	<b>3.754E-02</b>

Colour Checker:

As can be seen from **Tables 9-10**, the MFO-GWO-RRVFL algorithm still achieved the best results in the two indexes of angle error and chromaticity error (Mean and Median). In terms of angle error, the MFO-GWO-RRVFL algorithm only got sixth place in the Best25 and got second place in the Worst25. The results of the algorithm in Best25 were not very satisfactory, which is 0.157 different from the best result. In terms of chromaticity error, the MFO-GWO-RRVFL algorithm got second place in Best25 and got first place in Worst25. Among them, the results of the MFO-GWO-RRVFL algorithm were 6.31E-04 different from the best result.

**Table 9.** Angle errors comparisons on Colour Checker

Algorithm	Mean	Median	Best25	Worst25
MFO-GWO-RRVFL	<b>1.797</b>	<b>1.428</b>	0.610	3.606
GWO-RRVFL	2.991	2.508	1.067	5.541
MFO-RRVFL	1.812	1.499	0.585	<b>3.509</b>
ABC-RRVFL	4.367	3.276	0.973	9.772
BSO-RRVFL	2.184	1.779	<b>0.453</b>	4.447
PSO-RRVFL	3.947	3.026	1.421	7.853
RRVFL	2.292	2.229	0.634	4.346
RELM	2.433	1.691	0.766	5.121
BP	2.634	2.266	0.552	5.298
SVR	2.269	1.731	0.514	4.820

**Table 10.** Chromaticity errors comparisons on Colour Checker

Algorithm	Mean	Median	Best25	Worst25
MFO-GWO-RRVFL	<b>1.276E-02</b>	<b>9.781E-03</b>	4.633E-03	<b>2.464E-02</b>
GWO-RRVFL	2.190E-02	2.003E-02	8.221E-03	3.916E-02
MFO-RRVFL	1.300E-02	1.040E-02	4.695E-03	2.530E-02
ABC-RRVFL	3.426E-02	2.696E-02	8.264E-03	7.594E-02
BSO-RRVFL	1.665E-02	1.429E-02	4.002E-03	3.261E-02
PSO-RRVFL	2.959E-02	2.500E-02	9.941E-03	5.980E-02
RRVFL	1.700E-02	1.539E-02	4.948E-03	3.196E-02
RELM	1.794E-02	1.362E-02	6.121E-03	3.662E-02
BP	2.020E-02	1.581E-02	4.485E-03	4.195E-02
SVR	1.703E-02	1.338E-02	<b>4.160E-03</b>	3.620E-02

SFU Grey-ball:

The experimental data in **Table 11 and Table 12** show that the MFO-GWO-RRVFL achieves the best performance in Mean, Medium, Best25 and Worst25 of angle error and chromaticity error. This result shows that the MFO-GWO-RRVFL proposed in this paper has

made some improvement in the lighting correction effect after using a large volume data set to avoid some accidental experimental results.

**Table 11.** Angle errors comparisons on SFU Grey-ball (minimum values are displayed in bold)

Algorithm	Mean	Median	Best25	Worst25
MFO-GWO-RRVFL	<b>6.014</b>	<b>4.209</b>	<b>1.925</b>	<b>8.700</b>
GWO-RRVFL	6.846	5.743	2.906	9.890
MFO-RRVFL	7.389	6.239	3.421	10.784
ABC-RRVFL	7.235	6.089	3.094	10.499
BSO-RRVFL	7.101	6.133	3.123	10.282
PSO-RRVFL	7.098	6.189	3.042	10.126
RRVFL	7.417	6.189	3.367	10.948
RELM	7.586	6.640	3.695	10.902
BP	8.763	7.562	5.245	12.273
SVR	7.506	6.566	3.656	10.752

**Table 12.** Chromaticity error comparisons on SFU Grey-ball (minimum value is displayed in bold)

Algorithm	Mean	Median	Best25	Worst25
MFO-GWO-RRVFL	<b>4.361E-02</b>	<b>2.982E-02</b>	<b>1.411E-02</b>	<b>6.340E-02</b>
GWO-RRVFL	4.909E-02	3.994E-02	2.072E-02	7.012E-02
MFO-RRVFL	5.319E-02	4.367E-02	2.483E-02	7.696E-02
ABC-RRVFL	5.241E-02	4.282E-02	2.199E-02	7.514E-02
BSO-RRVFL	5.135E-02	4.384E-02	2.257E-02	7.358E-02
PSO-RRVFL	5.103E-02	4.326E-02	2.236E-02	7.219E-02
RRVFL	5.325E-02	4.306E-02	2.389E-02	7.862E-02
RELM	5.459E-02	4.616E-02	2.589E-02	7.708E-02
BP	6.284E-02	5.319E-02	3.738E-02	8.572E-02
SVR	5.380E-02	4.569E-02	2.505E-02	7.763E-02

From the results of these three sets of experiments, we can see that the proposed method still has good correction performance, indicating that the method has good robustness.

## 6. Conclusion

In our paper, we proposed the MFO-GWO-RRVFL method to simultaneously optimise the input feature and weight, hidden bias and node of the RRVFL by implementing the MFO-GWO algorithm. The proposed method did not require human intervention to adjust the parameters. In addition, it exhibited a fast convergence speed and good searching effect. However, when applied to illumination correction, the proposed MFO-GWO-RRVFL algorithm exhibited a small angle error and chromaticity error.

## References

- [1] T. Li, S. Rong, X. Cao, Y. Liu, L. Chen, B. He, "Underwater image enhancement framework and its application on an autonomous underwater vehicle platform," *Opt.Eng.*, vol. 59, no. 8, pp. 083102, Aug. 2020. [Article \(CrossRef Link\)](#)
- [2] Z. Zhou, R. Xu, D. Wu, Z. Zhu, H. Wang, "Illumination correction of dyed fabrics approach using Bagging-based ensemble particle swarm optimization-extreme learning machine," *Opt.Eng.*, vol.55, no.9, pp. 093102, Sep. 2016. [Article \(CrossRef Link\)](#)
- [3] V. C. Cardei, B. Funt, K. Barnard, "Estimating the scene illumination chromaticity by using a neural network," *JOSA. A.*, vol.19, no.12, pp. 2374-2386, Dec. 2002. [Article \(CrossRef Link\)](#)

- [4] W. Xiong, B. Funt, "Estimating illumination chromaticity via support vector regression," *Journal of Imaging Science and Technology*, vol.50, no.4, pp. 341-348, Jul./Aug. 2006. [Article \(CrossRef Link\)](#)
- [5] B. Li, W. Xiong, D. Xu, H. Bao, "A supervised combination strategy for illumination chromaticity estimation," *ACM Transactions on Applied Perception*, vol.8, no.1, pp. 1-17, Oct. 2010. [Article \(CrossRef Link\)](#)
- [6] Z. Zhou, D. Liu, J. Guo, J. Zhang, Z. Zhu, C. Wang, "Dyed fabric illumination estimation with regularized random vector function link network," *Color Res Appl*, vol.46, no.2, pp.376-387, Apr. 2021. [Article \(CrossRef Link\)](#)
- [7] F. Kang, J. Li, H. Li, "Artificial bee colony algorithm and pattern search hybridized for global optimization," *Applied Soft Computing*, vol.13, no.4, pp.1781-1791, Apr. 2013. [Article \(CrossRef Link\)](#)
- [8] R. Thangaraj, M. Pant, A. Abraham, P. Bouvry, "Particle swarm optimization: Hybridization perspectives and experimental illustrations," *Applied Mathematics and Computation*, vol.217, no.12, pp. 5208-5226, Feb. 2011. [Article \(CrossRef Link\)](#)
- [9] S. Mirjalili, S.M. Mirjalili, A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw*, vol. 69, pp. 46-61, Mar. 2014. [Article \(CrossRef Link\)](#)
- [10] S. Cheng, Q. Qin, J. Chen, Y. Shi, "Brain storm optimization algorithm: a review," *Artificial Intelligence Review*, vol.46, no. 4, pp. 445-458, Dec. 2016. [Article \(CrossRef Link\)](#)
- [11] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowledge-based Systems*, vol.89, pp. 228-249, Nov. 2015. [Article \(CrossRef Link\)](#)
- [12] C. Wang, Z. Zhu, S. Chen, J. Yang, "Illumination correction via support vector regression based on improved whale optimization algorithm," *Color Res Appl*, vol. 46 no. 2, pp. 303-318, Apr. 2021. [Article \(CrossRef Link\)](#)
- [13] X. Liu, D. Yang, "Color constancy computation for dyed fabrics via improved marine predators algorithm optimized random vector functional-link network," *Color Res Appl*, vol.46, no. 5, pp.1066-1078, Oct. 2021. [Article \(CrossRef Link\)](#)
- [14] Z. Zhou, D. Liu, J. Zhang, Z. Zhu, D. Yang, L. Jiang, "Color difference classification for dyed fabrics based on differential evolution with dynamic parameter selection to optimize the output regularization extreme learning machine," *Fibres & Textiles in Eastern Europe*, vol.29, no.3, pp. 97-102, May 2021. [Article \(CrossRef Link\)](#)
- [15] F. Han, H. Yao, Q. Ling, "An improved evolutionary extreme learning machine based on particle swarm optimization," *Neurocomputing*, vol.116, pp. 87-93, Sep. 2013. [Article \(CrossRef Link\)](#)
- [16] M. Eshtay, H. Faris, N. Obeid, "Metaheuristic-based extreme learning machines: a review of design formulations and applications," *Int J Mach Learn Cybern*, vol.10, no. 6, pp.1543-1561, Jun. 2019. [Article \(CrossRef Link\)](#)
- [17] J. Li, W. Shi, D. Yang, "Color difference classification of dyed fabrics via a kernel extreme learning machine based on an improved grasshopper optimization algorithm," *Color Research and Application*, vol. 46, no. 2, pp. 388-401, Apr. 2021. [Article \(CrossRef Link\)](#)
- [18] X. Liu, D. Yang, "Color constancy computation for dyed fabrics via improved marine predators algorithm optimized random vector functional-link network," *Color Research and Application*, vol.46, no. 5, pp. 1066-1078, Oct. 2021. [Article \(CrossRef Link\)](#)
- [19] J.Li, W.Shi, D.Yang, "Fabric wrinkle evaluation model with regularized extreme learning machine based on improved Harris Hawks optimization," *Journal of the Textile Institute*, vol.113, no. 2, pp.199-211, Feb. 2022. [Article \(CrossRef Link\)](#)
- [20] Fedor V. Fomin, Petteri Kaski, "Exact exponential algorithms," *Commun. ACM*, vol.56, no.3, pp.80-88, March 2013. [Article \(CrossRef Link\)](#)
- [21] S. Fateme Attar, Mohammad Mohammadi, Seyed Hamid Reza Pasandideh, Bahman Naderi, "Formulation and exact algorithms for electric vehicle production routing problem," *Expert Systems with Applications*, vol.204, April 2022. [Article \(CrossRef Link\)](#)
- [22] Expósito-Izquierdo, C., López-Plata, I. and Moreno-Vega, J.M, "Problem MetaHeuristic Solver: An educational tool aimed at studying heuristic optimization methods," *Comput Appl Eng Educ*, vol.23, no.6, pp.897-909, June 2015. [Article \(CrossRef Link\)](#)

- [23] Kate Smith, M. Palaniswami, M. Krishnamoorthy, "Traditional heuristic versus Hopfield neural network approaches to a car sequencing problem," *European Journal of Operational Research*, vol.93, no.2, pp.300-316, September 1996. [Article \(CrossRef Link\)](#)
- [24] Alexandros Tsipianitis, Yiannis Tsompanakis, "Optimizing the seismic response of base-isolated liquid storage tanks using swarm intelligence algorithms," *Computers & Structures*, vol.243, January 2021. [Article \(CrossRef Link\)](#)
- [25] H. R. Tizhoosh, "Opposition-Based Learning: A New Scheme for Machine Intelligence," in *Proc. of International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, pp. 695-701, November 2005. [Article \(CrossRef Link\)](#)
- [26] Yu, Y., Gao, S., Cheng, S. et al, "CBSO: a memetic brain storm optimization with chaotic local search," *Memetic Comp*, vol.10, pp.357-367, December 2017. [Article \(CrossRef Link\)](#)
- [27] Foster, David H, "Color constancy," *Vision research*, vol.51, no.7, pp.674-700, April 2011. [Article \(CrossRef Link\)](#)
- [28] Celik, Turgay, and Tardi Tjahjadi, "Adaptive colour constancy algorithm using discrete wavelet transform," *Computer Vision and Image Understanding*, vol.116, no. 4, pp.561-571, April 2012. [Article \(CrossRef Link\)](#)
- [29] Muniraj, Manigandan, and Vaithiyanathan Dhandapani, "Underwater image enhancement by color correction and color constancy via Retinex for detail preserving," *Computers and Electrical Engineering*, vol.100, May 2022. [Article \(CrossRef Link\)](#)
- [30] Finlayson, Graham D., Mark S. Drew, and Brian V. Funt, "Color constancy: generalized diagonal transforms suffice," *JOSA A*, vol.11, no.11, pp. 3011-3019, November 1994. [Article \(CrossRef Link\)](#)
- [31] Gijssenij, Arjan, and Theo Gevers, "Color constancy using natural image statistics," in *Proc. of 2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp.1-8, June 2007. [Article \(CrossRef Link\)](#)
- [32] Gijssenij, Arjan, and Theo Gevers, "Color constancy using natural image statistics and scene semantics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.33, no.4, pp.687-698, May 2011. [Article \(CrossRef Link\)](#)
- [33] Oh, Seoung Wug, and Seon Joo Kim, "Approaching the computational color constancy as a classification problem through deep learning," *Pattern Recognition*, vol. 61, pp. 405-416, January 2017. [Article \(CrossRef Link\)](#)
- [34] Lou, Zhongyu, Theo Gevers, Ninghang Hu, and Marcel P. Lucassen, "Color Constancy by Deep Learning," in *Proc. of BMVC*, pp.76.1-76.12, September 2015. [Article \(CrossRef Link\)](#)
- [35] Das, Partha, Anil S. Baslamisli, Yang Liu, Sezer Karaoglu, and Theo Gevers, "Color constancy by GANs: An experimental survey," *arXiv:1812.03085*, 2018. [Article \(CrossRef Link\)](#)
- [36] Xiao, Jin, Shuhang Gu, and Lei Zhang, "Multi-domain learning for accurate and few-shot color constancy," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3258-3267, 2020. [Article \(CrossRef Link\)](#)
- [37] Y. H. Pao, G. H. Park, D. J. Sobajic, "Learning and generalization characteristics of the random vector functional-link net," *Neurocomputing*, vol.6, no.2, pp.163-180, Apr. 1994. [Article \(CrossRef Link\)](#)
- [38] K. Barnard, V. Cardei and B. Funt, "A comparison of computational color constancy algorithms. I: Methodology and experiments with synthesized data," *IEEE Transactions on Image Processing*, vol.11, no.9, pp.972-984, Sep. 2002. [Article \(CrossRef Link\)](#)
- [39] E. Ershov et al., "The Cube++ Illumination Estimation Dataset," *IEEE Access*, vol. 8, pp.227511-227527, Dec. 2020. [Article \(CrossRef Link\)](#)
- [40] L. Shi and B. Funt, "Re-processed Version of the Gehler Color Constancy Dataset of 568 Images," [Online]. Available: <https://www2.cs.sfu.ca/~colour/data/>
- [41] J.Yang, M.Cai and Z.Zhou, "Evolving convolution neural network by optimal regularization random vector functional link for computational color constancy," *Optical Engineering*, vol.61, no.10, pp.103102, Oct. 2022. [Article \(CrossRef Link\)](#)



**Xiaochun Zhang** is a lecturer at Wenzhou Polytechnic and she is mainly engaged in machine vision.



**Zhiyu Zhou** received his master and Ph.D. degrees from Zhejiang Sci-Tech University in 2004 and 2018, respectively. Currently, he is an associate professor at Zhejiang Sci-Tech University and is mainly engaged in computer vision, machine learning, and robotic tracking