

Research Article A Novel Real-Time Image Restoration Algorithm in Edge Computing

Xingmin Ma,¹ Shenggang Xu,¹ Fengping An ^(b),² and Fuhong Lin ^(b)

¹School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China ²Huaiyin Normal University, Huai'an 223001, China

Correspondence should be addressed to Fuhong Lin; fhlin@ustb.edu.cn

Received 18 April 2018; Accepted 9 July 2018; Published 9 August 2018

Academic Editor: Ao Zhou

Copyright © 2018 Xingmin Ma et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Owning to the high processing complexity, the image restoration can only be processed offline and hardly be applied in the realtime production life. The development of edge computing provides a new solution for real-time image restoration. It can upload the original image to the edge node to process in real time and then return results to users immediately. However, the processing capacity of the edge node is still limited which requires a lightweight image restoration algorithm. A novel real-time image restoration algorithm is proposed in edge computing. Firstly, 10 classical functions are used to determine the population size and maximum iteration times of traction fruit fly optimization algorithm (TFOA). Secondly, TFOA is used to optimize the optimal parameters of least squares support vector regression (LSSVR) kernel function, and the error function of image restoration process, the training process is to establish a mapping relationship between the degraded image and the adjacent pixels of the original image. The relationship is established; the degraded image can be restored by using the mapping relationship. Through the comparison and analysis of experiments, the proposed method can meet the requirements of real-time image restoration, and the proposed algorithm can speed up the image restoration and improve the image quality.

1. Introduction

With the development of science and technology, more and more users use their mobile phones for photography. During imaging of the digital image, it is easy to produce image blur due to jitter, light, and many other factors [1], and this phenomenon is called image degradation. Therefore, an effective parameter setting needs to be performed in image restoration processing, which consumes a large amount of resources. The current solution is to use offline servers and other types of processing equipment, which cannot be done in real time. The emergence of edge computing makes real-time processing of such applications possible. Edge computing can meet the user's real-time computing service requests and provides users with low-latency image processing services. The local image is uploaded to edge nodes for real-time processing. We can deploy the Traction Fruit fly Optimization Algorithm (TFOA) and Least Squares Support Vector Regression (LLSVR) to edge nodes. However,

the processing capacity of the edge node is limited, so it is necessary to propose a lightweight image restoration algorithm. In this way, the photos taken by the mobile phone can be processed in real time in the edge node.

Image restoration is the reverse process of image degradation. And its ultimate goal is to remove blur and interference, so that the image is restored as close as possible to the original one [2]. One of the most typical image blur models is Gaussian blur. Gaussian point spread functions are common in many image acquisition, measurement, and transmission systems. So, many degradation processes can be approximated by Gaussian blur models. The study of Gaussian blurred image restoration is of great significance to other types of degraded image restoration [3]. Most of the traditional restoration methods are deconvolution processes, such as inverse filtering, wiener filtering, least square method, and Lucy-Richardson (LR), which are commonly referred to as nonblind restoration methods [4, 5]. These methods all need to know the point spread function of imaging system,



FIGURE 1: Population iterative search for food in fruit fly.

which is very difficult in practical application. In contrast, blind restoration method uses the information carried by the image itself or the degradation model estimation under the condition that the point spread function is unknown and combines the additional conditions to solve the problem [6]. This kind of method is widely used at present, but the result is not stable, and the calculation is large and the convergence is slow.

With the rise of in-depth learning in recent years, various machine intelligence models, such as BP neural network and support vector machine, have been widely used in the field of image restoration. In 2011, Mukherjee et al. used a Multilayered Quantum Backpropagation Neural Network for Image Restoration [7]; Sethy et al. proposed an image restoration method based on BP neural network, which has achieved good results, but there is still room for improvement [8]. References [9, 10] use fuzzy neural network model for image restoration and the image quality has been improved to some extent, but there is still room for optimization. Reference [11] proposes image restoration based on support vector regression (SVR), which improve convergence accuracy and achieves better restoration effect. However, in the optimization process of SVM parameters using FOA, the two internal parameters are not optimized, and only empirical values are given. Based on the existing work, this paper proposes an image restoration method based on TFOA-LSSVR. Through the comparison and analysis of experiments, the proposed method can meet the requirements of real-time image restoration.

2. Background Materials

2.1. Fruit Fly Optimization Algorithm (FOA). FOA is an algorithm for finding food sources based on fruit fly, which is a globally optimized algorithm [12]. The olfactory and visual

organs of fruit fly are very developed. It first collects the smell of food while foraging, the position where food and others companions congregate can also be visually detected after approaching the food location and fly in this direction. A schematic diagram of the iterative foraging process for fruit fly populations is shown in Figure 1. The algorithm combines the global search of population with individual information exchange. On that basis of a global search, global optimal solution is updated through individual information exchange. And finally the algorithm is terminated under the condition that either the maximum iteration number or the convergence target precision is met. The steps of FOA are as follows.

Step 1. The population size is expressed by sizepop and the maximum iteration number is expressed by max gen. The random initial position of fruit fly is expressed by X_axis, Y_axis .

Step 2. Give fruit fly individuals a random direction and distance to use their sense of smell to search for food; *StepLength* is the search distance, namely, the iterative step value; that is,

$$X(i) = X_axis + StepLength$$
(1)

$$Y(i) = Y_axis + StepLength$$
(2)

Step 3. Since the initial position of the food is unknown, we first estimate the distance $Dist_i$ between individual and the origin and then set the smell concentration S_i of individual, which we define here as the reciprocal of the distance.

$$Dist_i = \sqrt{X_i^2 + Y_i^2} \tag{3}$$

$$S_i = \frac{1}{Dist_i} \tag{4}$$

Step 4. The smell concentration value S_i brought into the smell concentration determination function to calculate the smell concentration *Smell*_i;

$$Smell_i = Function(S_i)$$
 (5)

Step 5. Find the individual with the best odor concentration in the population;

$$[bestsmell, bestindex] = \min(bestsmell)$$
(6)

Step 6. The optimum smell concentration value of bestsmell and coordinates X, Y are recorded and retained; this time, the fruit fly population uses vision to fly to that location;

$$X_axis = X (bestindex)$$
(7)

$$Y_axis = Y (bestindex)$$
(8)

Step 7. Enter iteration optimization, repeatedly execute Steps 2 to 5, and judge whether the optimal taste concentration is better than the optimal smell concentration of the previous iteration or not and the current iteration number is less than the maximum iteration number max gen; if yes, Step 6 is executed.

This paper introduces a new FOA, which is TFOA [13]. The optimization algorithm has no change on the two main parameters. Instead, the direction of iteration is strategically optimized, and it solves the problem of falling into a local optimal solution in the solving process. The results of each iteration are recorded. When the algorithm is locally optimal, optimalpop represents the best individual [13, 14], and badPop represents the worst individual in each iteration. The size of the Drosophila population is represented by sizepop. The final direction of the iteration is represented by the following formula:

direction

$$= \begin{cases} 1 & optimalpop^{j} > badPop_{i}^{j} \\ -1 & optimalpop^{j} < badPop_{i}^{j} \\ RandomDirection & optimalpop^{j} = badPop_{i}^{j} \end{cases}$$
(10)

The new individual generation method is as shown in (11).

$$individual_{i}^{m} = optimalpop^{m}$$

$$+ (randius * RandDirection)$$
(11)

$$individual_{0} = rand (1, \dim ension)$$
$$individual_{i} = individual_{i-1} * (E - individual_{i-1})$$
(12)
$$* 4$$

In (12), the entire vector is 1.

The main steps of the TFOA [13] are as follows.

1 / 1 1.

Step 1. The Drosophila population is produced according to (11), that adaptability of the Drosophila individual is calculated, and the worst individual is recorded.

Step 2. The optimal smell concentration and initial position of the population were changed. In this process, if the smell concentration is consistent, the iteration is continued and the count is increased.

Step 3. During the iteration, if the algorithm ends prematurely, (11) is executed to jump out of local optimization. When a better fruit fly is discovered, the population's location is updated and the search continues around it. If it ends too quickly, expand the search radius and continue iterative optimization. When the traction operation is complete, the worst person is rerecorded in each iteration process.

Step 4. Steps 1 to 3 are repeatedly executed, and whether the optimal taste concentration is better than the optimal taste concentration of the previous iteration is judged, until meeting the requirements of the algorithm.

2.2. Support Vector Machine (SVM). SVM is developed from statistical theory [15]. LSSVR is developed from SVM through special treatment [11]; LSSVR is suitable for regression model of static nonlinear function estimation. SVM algorithm has been applied to classification problems since it was proposed; now in the classification direction is very mature, the classification method is effectively extended, and the current extension is also very effective in fitting curves and nonlinear regression estimation.

Trained array $\{(x_i, y_i), i = 1, 2, \dots, l\}$, the *i*'th value in this array combination, is representing $x_i \in \mathbb{R}^d$, it is known that there are vectors $x_i = (x_i^1, x_i^2, \dots, x_i^d)$ of d dimensions at the same time, and that output result is set to $y_i \in R$. The missing parameter ε is

$$y - f(x)|_{\varepsilon} = \begin{cases} 0 & |y - f(x)| \le \varepsilon \\ |y - f(x)| - \varepsilon & |y - f(x)| > \varepsilon \end{cases}$$
(13)

When the loss parameter is 0, assuming that the difference between the estimated value and the actual value of the final result obtained by training the f(x) into the support vector machine is smaller than the lost parameter, finding a proper kernel function $k(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j)$ and mapping the kernel function $k(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j)$ to a linear highdimensional space for regression prediction, assume here that the function is

$$f(x) = \omega \cdot \varphi(x) + b \tag{14}$$

In formula (14), $\omega \in \mathbb{R}^d$ is expressed as the weight value of the vector, $b \in R$ represents the threshold calculated for the data, and (\cdot) is expressed as the inner product operation of the data. The value of b and ω is finally obtained. Error function under the premise that meets $\omega^T \omega/2$ minimizes ε , ξ_i, ξ_i^* are added to the optimization calculation, and the final expression is

$$\min_{w,b,\xi_i,\xi_i^*} \quad \frac{\|w\|^2}{2} + C \sum_{i=1}^l \left(\xi_i + \xi_i^*\right) \tag{15}$$

s.t.
$$y_i - (w \cdot \varphi(x_i) + b) \le \varepsilon + \xi_i$$
 (16)

$$(w \cdot \varphi(x_i) + b) - y_i \le \varepsilon + \xi_i^*$$
(17)

$$\xi_i, \xi_i^* \ge 0, \ i = 1, 2, \cdots, l$$
 (18)

$$\max \left[-\frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} (a_i - a_i^*) (a_j - a_j^*) k(x_i, x_j) - \varepsilon \sum_{i=1}^{l} (a_i + a_i^*) + \sum_{i=1}^{l} yi (a_i - a_i^*) \right]$$
(19)

s.t.
$$\sum_{i=1}^{l} (a_i - a_i^*) = 0$$
 (20)

$$0 \le a_i, a_i^* \le C \tag{21}$$

It is assumed here that the penalty parameter *C* is a normal number. The above problems are solved optimally, support vector $(a_i - a_i^*) \neq 0$, and $w = \sum_{i=1}^{l} (a_i - a_i^*) \varphi(x_i)$ Then, there is

$$b = \frac{1}{N_{Nsv}} \left\{ \sum_{0 \le a_i \le C} \left[y_i - \sum_{x_j \in sv} \left(a_j - a_j^* \right) k\left(x_j, x_i \right) - \varepsilon \right] + \sum_{0 \le a_i^* \le C} \left[y_i - \sum_{x_j \in sv} \left(a_j - a_j^* \right) k\left(x_j, x_i \right) - \varepsilon \right] + \varepsilon \right] \right\}$$

$$(22)$$

The numb of standard support vectors in (22) is N_{Nsv} ; obtain a *b* threshold and a regression function:

$$f(x) = \sum_{x_i \in s_{\nu}} (a_i - a_i^*) k(x_i, x) + b$$
(23)

There are three kernel functions commonly used in support vector machines:

First: kernel function polynomial

$$K\left(x_{i}, x_{j}\right) = \left[\left(x_{i} \cdot x_{j}'\right) + 1\right]^{q}$$
(24)

The coefficient of the polynomial in (24) is *q*, the value is determined by the application itself, and the parameter is adjusted according to the actual situation.

Second, the radial basis function (RBF kernel function)

$$K\left(x_{i}, x_{j}\right) = \exp\left\{-\frac{\left|x_{i} - x_{j}\right|^{2}}{2\sigma^{2}}\right\}$$
(25)

The core parameter in (25) is σ ; it uses Gaussian kernel function of mean square error; the larger the value of σ , the wider the width of the Gaussian function.

Third, Sigmoid kernel function is as follows:

$$K\left(x_{i}, x_{j}\right) = \frac{1}{\left(1 + \exp\left(-\left|x_{i} - x_{j}\right|\right)\right)}$$
(26)

The LSSVR represents the following:

$$f(X) = \sum_{i=1}^{n} a_i K(x, x') + b$$
 (27)

$$K\left(x_{i}, x_{j}\right) = \exp\left\{-\frac{\left|x_{i} - x_{j}\right|^{2}}{2\sigma^{2}}\right\}$$
(28)

We need to optimize the selection of parameter *C* and δ before conducting sample training.

3. The Proposed Methodology

3.1. Process of Image Restoration Processing. Assuming that the original image is f(x, y), the degraded image may be expressed as

$$g(x, y) = H[f(x, y)] + \eta(x, y)$$
⁽²⁹⁾

In (29), *H* denotes a degraded image, and $\eta(x, y)$ denotes additive noise.

If H is a linear, spatially invariant process, the degraded image may be represented in the spatial frequency domain as

$$g(x, y) = f(x, y) * h(x, y) + \eta(x, y)$$
(30)

In (30), * is a convolution operation, f(x, y), h(x, y), $\eta(x, y)$ are original image, degraded image, and additive noise. Image restoration is a parameter estimation process. When the fuzzy kernel function is Gaussian, the degeneration process is called Gaussian blur. This type of blurred image can be seen as a result of convolution operation between the convolution kernel of Gaussian distribution and the clear image. Namely, that pixel value of a certain point of the image

and the surrounding pixel value are weighted and averaged according to the weight of the Gaussian distribution to obtain the pixel value after the position is degraded.

3.2. Optimization Algorithm of TFOA Based on LSSVR. In this paper, TFOA-LSSVR is introduced into the mapping relationship model between images. In the process of image restoration, the selection criteria of the original image and the blurred image are consistent. Assuming that the size of the image is $m \times n$, a i - 1 window of an original image corresponds to the corresponding center pixel in the process of constructing a training sample. If there is a degraded image completely recorded by the near window, we can get $(m - i + 1) \times (n - i + 1)$ support vectors in the process of degradation, and vector dimension is $i^2 \times 1$. We represents this vector with X_i and the pixel i with Y_i , then image restoration steps are as follows [12].

Step 1. We initialize the TFOA.

Step 2. We set up a two-dimensional space. According to Section 2.1, we have the following formula:

$$D(i, 1) = \sqrt{X(i, 1)^{2} + Y(i, 1)^{2}}$$

$$D(i, 2) = \sqrt{X(i, 2)^{2} + Y(i, 2)^{2}}$$

$$S(i, 1) = \frac{1}{D(i, 1)}$$

$$S(i, 2) = \frac{1}{D(i, 2)}$$
(31)
(32)

Step 3. The mean square error (MSE) function is used as the odor concentration function of TFOA when a mapping relationship is established between the original image and the degraded image.

$$MSE = \frac{\sum_{i=1}^{mn} \left(\hat{t} - t\right)^2}{mn}$$
(33)

In (33), \hat{t} represents the original image, t represents a restored image, the size of the image is $m \times n$, and the parameter $[C, \delta]$ in the least squares support vector machine is represented by the parameter [S, (i, 1), S(i, 2)].

Step 4. The odor concentration value is calculated by comparing the calculated minimum odor concentration and bringing this value into the LSSVR model, while preserving the optimal location.

Step 5. Carrying out iteration optimization, repeating Steps 2-4, and stopping iteration optimization when the iteration number reaches the maximum, the parameters [S, (i, 1), S(i, 2)] are obtained, and then $[C, \delta]$ are obtained.

After the training sample is completed, the corresponding picture attribute is created by using the same picture size when a new degraded image needs to be restored. Through the LSSVR model, the pixel value of the image is predicted, and finally the pixel value of the image is displayed by pixels.

4. Experiments and Applications

4.1. Parameter Optimization Analysis of TFOA. The server was simulated in Windows 7 operating system (i5-3210M, 2.5 GHz CPU, 8.00 GB), and the TFOA algorithm was implemented using Matlab2014a. The simulation experiment is carried out by taking the minimum value of 10 benchmark functions [14, 16] as an example. The effects of parameters on the convergence accuracy and speed of the algorithm were analyzed and compared. Finally, the appropriate *sizepop* and max *gen* were determined. Name, function form, search interval, and function optimal value of 10 benchmark functions are in Table 1. The picture material used in this paper comes from [17].

4.1.1. Effect of Population Size on Algorithm Performance. In the experiment, we use progressive approaches to population size sizepop. The final experimental results were analyzed to find out the relationship among the population size, the convergence accuracy, and time complexity of the algorithm. The parameters were set as follows: population size separate values (5,10,15,20); max *gen* is 150 times; optimization iterative step is valued by TFOA algorithm. The search interval is shown in Table 1. The optimal mean value, average running time, and relative change rate of convergence accuracy of TFOA global optimization function are taken as evaluation indexes. The experimental results of 10 test functions after 50 consecutive runs are shown in Table 2.

In Table 2, optimized mean represents the arithmetic mean of the minimum value of the global optimization function; it reflects the convergence accuracy of the algorithm: the smaller the value, the higher the convergence accuracy of the algorithm. Average operation time is the arithmetic average of the algorithm's run time, which is the average time required for the algorithm to run once. Convergence condition refers to optimal mean corresponding to low population sizeoptimal mean corresponding to high population size/optimal mean corresponding to low population size. It shows that the convergence accuracy of the algorithm increases with the increase of population size, and the larger the value, the greater the convergence accuracy of the algorithm. At the same time, as we can see from Table 2, with the increase of sizepop, the convergence accuracy of the algorithm is improved, and the average running time is proportional to the increase; however, with the increase of population size, the relative rate of change of convergence accuracy shows a downward trend. The trend line of optimized mean of 10 test functions with increasing population size is shown in Figure 2. In the figure, the ordinate is expressed by optimized mean and the abscissa is population size.

As we can see from Figure 2, at the initial stage of population size increase, the optimized mean decreases monotonically with the increase of population size, and the relative rate of change is the largest; that is, the convergence accuracy of the algorithm is the largest. In the middle stage of population size increase, the optimized mean continues to decrease monotonically with increasing population size, but the relative rate of change decreases; at the later stage of population increase, on functions F4, F5, and F8, optimized

Function ID	Function Name	Equation	Function Type	Dimension	Bounds of X
F1	Classic	$f(x,y) = -x\cos(2yy) - y\sin(2yx)$	Multimodal	2	UB(2)
		$f_1(x,y) = x \cos(2py) - y \sin(2px)$			LB(-2)
F2	Bohachevsky	$f_2(x, y) =$	Multimodal	2	UB(10)
		$x^{2}+y^{2}-0.3 * \cos(3\pi x)+0.3 * \cos(4\pi y)+0.3$			LB(-10)
F3	Step	$f(x) = \sum_{n=1}^{n} (x + 0.5)^2$	Unimodal	30	UB(100)
		$J_{3}(x) - \sum_{i=1}^{2} ([x_{i} + 0.5])$			LB(-100)
F4	Easom	$(x-\pi)^2 - (y-\pi)^2$	Multimodal	2	UB(10)
		$f_4(x) = -\cos x \times \cos y \times e^{-\cos x}$			LB(-10)
F5	Sphere	$f(x) = \sum_{n=1}^{n} x^2$	Unimodal	30	UB(100)
		$\int_{5} (x) - \sum_{i=1}^{2} x_{i}$			LB(-100)
E6	Rastrigin	$f(x) = \sum_{n=1}^{n} (x^2 - 10 \cos(2\pi x) + 10)$	Multimodal	30	UB(5.12)
10		$\int_{6} (x) - \sum_{i=1} (x_{i} - 10\cos(2\pi x_{i}) + 10)$			LB(-5.12)
F7	Quartic	$f_7(x) = \sum_{i=1}^n ix_i^4 + rand()$	Unimodal	30	UB(1.28)
					LB(-1.28)
F8	Sum squares	$f(x) = \sum_{n=1}^{n} i x^2$	Unimodal	30	UB(100)
		$J_8(x) - \sum_{i=1}^{i} ix_i$			LB(-100)
F9	Ackley	$f_{-1}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{2} \sum_{n=1}^{n} x^2}\right)$	Multimodal	30	UB(32)
		$-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$			00(02)
					LB(-32)
E10	Griewank	$f_{i}(x) = \frac{1}{2} \sum_{i=1}^{n} \frac{1}{2} \sum_{i=$	Multimodal	30	UB(100)
F10		$J_{10}(x) = \frac{1}{4000} \sum_{i=1}^{2} x_i - \prod_{i=1}^{2} \cos\left(\frac{1}{\sqrt{i}}\right) + 1$			LB(-100)

TABLE 1: The classical test functions.

mean continues to decrease with increasing population size, but the relative rate of change is also smaller. Therefore, the synthesis of the above analysis can draw the following two conclusions: Firstly, with the increase of population size, the time cost of the algorithm also increases. Secondly, with the increase of population size, the convergence accuracy of the algorithm is improved; however, with the increase of population size, the convergence accuracy of the algorithm cannot be improved significantly, because it is easy to fall into local optimum at the later stage of algorithm iteration. Therefore, the appropriate population size can find the best balance between algorithm performance and running time. In this paper, 15 is chosen as the population size of TFOA.

4.1.2. Influence of Maximum Iteration Number on Algorithm Performance. In the experiment, we use progressive approaches to population size max *gen*; the parameters were set as follows: max *gen* separate values are 10, 50, 150, 500, and 1000, *sizepop* is 15, and the search interval is shown in Table 1. The optimal mean value obtained from the minimum value of the global optimization function and the running times of the algorithm are taken as evaluation indexes. The experimental results of 10 test functions after 50 consecutive runs are shown in Table 3, the larger max *gen*, the higher the accuracy of the optimized mean and the higher the average running time. The trend line of the optimized mean values of the 10 test

functions as max *gen* increases is shown in Figure 3, with the ordinate indicated by optimized mean, and the abscissa is maximum iteration number.

In Figure 3, we can draw the following two conclusions: Firstly, increasing the max *gen* will inevitably increase the computation time consumed by the algorithm. Secondly, the convergence accuracy of the algorithm can be improved by increasing the max *gen*, but that continuous increase of the maximum iteration numb does not lead to the continuous and obvious improvement of the convergence precision of the algorithm; this is because it is easy to fall into local optimum at the later stage of algorithm iteration. Therefore, the appropriate maximum number of iterations can find the best balance between algorithm performance and runtime. The max *gen* used in this paper is 500.

4.2. Image Restoration Analysis of LSSVM- TFOA. The server was simulated in edge computing. In order to make the algorithm convincing, we add different types of blur and noise to the standard image for testing. In the experiment, we use the size 512×512 for both the training image and the test image through normalization processing. Use the optimization results in the previous section that *sizepop* is 15 and max *gen* is 500. The initial position of Drosophila is random. In order to evaluate the image restoration effect quantitatively, the peak signal-to-noise ratio (PSNR) and

Function	Evolution in day	sizepop				
	Evaluation index	5	10	15	20	
F1	Optimized mean	4.450	4.398	4.381	4.380	
	Average operating time (s)	0.617	0.953	1.385	1.671	
	Convergence condition	-	1.1709	0.3843	0.0137	
	Optimized mean	8.6098	8.6059	8.6046	8.6045	
F2	Average operating time (s)	0.543	0.969	1.421	1.606	
	Convergence condition	-	0.0453	0.0151	0.0012	
	Optimized mean	0.002541	0.002436	0.002412	0.002408	
F3	Average operating time (s)	0.586	0.849	1.287	1.599	
	Convergence condition	-	4.1322	0.9852	0.1658	
	Optimized mean	1.34003E-05	1.29E-05	1.28E-05	1.26E-05	
F4	Average operating time (s)	0.614	1.265	1.68	1.55	
	Convergence condition	-	3.5632	2.1095	0.8419	
	Optimized mean	0.0733	0.0409	0.03784	0.0301	
F5	Average operating time (s)	0.726	1.1	1.233	1.807	
	Convergence condition	-	44.2019	7.4817	20.4545	
F6	Optimized mean	8.8113	6.5458	5.44	4.728	
	Average operating time (s)	0.76	0.985	1.233	1.586	
	Convergence condition	-	25.7113	16.8933	13.0882	
	Optimized mean	15.6435	15.1187	14.2415	14.1232	
F7	Average operating time (s)	0.579	0.863	1.321	1.592	
	Convergence condition	-	3.3547	5.8021	0.8307	
F8	Optimized mean	2.5632	2.1998	1.6826	1.4576	
	Average operating time (s)	0.564	0.886	1.253	1.605	
	Convergence condition	-	14.1776	23.5112	13.3722	
F9	Optimized mean	0.2044	0.2028	0.1475	0.1382	
	Average operating time (s)	0.603	1.061	1.302	1.793	
	Convergence condition	-	0.7828	27.2682	6.3051	
F10	Optimized mean	1.008	1.004	1.002	1	
	Average operating time (s)	0.652	1.705	1.72	2.95	
	Convergence condition		0.3968	0.1992	0.1996	

TABLE 2: Effect of population size on algorithm performance.

normalized mean square error (NMSE) are used to evaluate the image restoration performance.

Experiment 1 (establishing a mapping relation of image restoration). In order to establish the mapping relationship between original image and blurred image, we add Gaussian blur $(5 \times 5, \delta = 0.8)$ and Gaussian random noise (variance=0.01) to Barbara image. Blurred image is generated by convolution of original image and Gaussian filter. Barbara's image restoration model is constructed from degraded and original Barbara images. After 500 iterations of training, the optimization results for parameters *C* and δ are 5.2 and 0.6231. In order to verify the image restoration effect of the algorithm in this paper, the support vector regression [15] and BP neural network algorithm [10] are selected to compare, and the final image restoration effect is shown in Figure 4 and Table 4.

From the final training results, we can know that the mapping approach can heighten the image quality of blurred image. The PSNR of BP neural network algorithm is low, while the NMSE is higher, which is because its initial parameters need to be selected and corrected. SVR method is better than BP neural network in image restoration, but it takes more time in cross-validation of parameters. However, this method takes less time and has higher PSNR and lower NMSE, which shows better image restoration performance.

Experiment 2 (comparison of degraded image restoration using the trained mapping). The TFOA-LSSVR mapping model proposed in this paper is applied to different blurred images. In order to make the algorithm convincing, there are three blurred ways, namely, motion blur ($L = 20, \delta = 30$), disk blur (R = 10), and passivation blur. They are,



FIGURE 2: The relationship between population size and optimized mean of F1-F10.



FIGURE 3: The relationship between max gen and optimized mean of F1-F10.

Function	Exclustion in day	maxgen					
	Evaluation index	10	50	150	500	1000	
F1	Optimized mean	4.4052	4.3775	4.3584	4.3456	4.3448	
	Average operating time (s)	0.311	0.943	1.771	6.14	11.16	
F2	Optimized mean	8.6921	8.6172	8.602	8.6002	8.6001	
	Average operating time (s)	0.275	0.658	1.751	5.871	10.983	
F3	Optimized mean	0.002637	0.002511	0.002412	0.002409	0.002406	
	Average operating time (s)	0.283	1.078	1.805	5.399	11.393	
F4	Optimized mean	1.88E-05	1.35E-05	1.24E-05	1.20E-05	1.18E-05	
	Average operating time (s)	0.246	0.662	1.861	5.345	11.085	
F5	Optimized mean	0.5993	0.0533	0.022	0.002	0.000858	
	Average operating time (s)	0.366	0.872	1.755	5.382	11.543	
EC.	Optimized mean	8.3914	8.0295	7.1533	7.12	7.1	
10	Average operating time (s)	0.351	0.704	1.733	5.417	11.641	
F7	Optimized mean	15.9877	15.7825	15.2618	14.9422	14.6644	
	Average operating time (s)	0.317	0.741	2.386	5.962	11.726	
F8	Optimized mean	2.304	2.0313	1.1926	1.0082	1.002	
	Average operating time (s)	0.248	0.684	1.798	5.516	11.231	
F9	Optimized mean	1.2648	0.2589	0.1368	0.04961	0.0209	
	Average operating time (s)	0.237	0.734	1.74	5.982	11.317	
F10	Optimized mean	1.013	1.011	1.008	1.0025	1.0022	
	Average operating time (s)	0.314	1.384	2.837	8.16	17.394	

TABLE 3: Effect of maximum iteration number on algorithm performance.

TABLE 4: Experimental results of different approach.

Approach	PSNR	NMSE	Time Cost
Blur image	18.1292	0.1495	-
BP neural network	23.3441	0.0914	13.2354
Support vector regression	26.9224	0.0758	20.1126
TFOA-LLSVR	28.6251	0.0652	9.1128

respectively, applied to three original standard images for blur processing simulation, and Gaussian noise is added to the motion blur images and disk blur images. In order to further explain the image restoration effect, Lucy-Richardson (LR) algorithm [18] is used to restore the different degraded image. The specific results are shown in Figure 5 and Table 5.

In this experiment, TFOA-LSSVR can improve image quality in PSNR and NMSE. But LR algorithm's recovery result is not very satisfactory. For motion blurred images added with Gaussian noise and disk blurred images added with Gaussian noise, the final image clarity of LR algorithm is not improved, and a large number of spots appear. For passivation blurred images, LR results showed some blurring and a large number of spots too. All TFOA-LSSVR algorithms recover images that are clearer than blurred images, and most of the noise is reduced. Therefore, the TFOA-LSSVR model in this paper has better application effect in image restoration.

5. Conclusion

This paper proposes an image restoration method in edge computing environment which achieves good image realtime restoration results. In this paper, TFOA and LSSVR are combined to establish a nonlinear mapping model for image restoration. Firstly, the population size and the maximum iteration number of TFOA are determined by using 10 test functions. Then, the LSSVR error function is optimized as an adaptive function of TFOA. The LSSVR with optimized parameters is used to establish the degradation relationship between the original image and the blurred image. Finally, the image restoration effect is verified. In order to verify the validity of the mapping method, the BP network and the SVR algorithm were compared, and the feasibility of the combination of TFOA and LSSVR was verified. At the same time, the nonlinear mapping model constructed by this algorithm can be used to recover the degenerate

Wireless Communications and Mobile Computing

TABLE 5: Comparison of different restored methods.

Blurs & Noise	Blur image		LR		TFOA-LLSVR	
Diurs & Noise	PSNR	NMSE	PSNR	NMSE	PSNR	NMSE
Motion Blur	19.2455	0.1489	19.3155	0.1326	24.3776	0.0689
Disk blur	17.9981	0.0986	18.3864	0.0997	22.1244	0.0891
Passivation blur	18.9987	0.0834	21.9166	0.0792	25.7311	0.0667



(a) Original Barbara image

(b) Blurred Barbara image

(c) Restored by BP neural network



(d) Restored by SVR



(e) Restored by TFOA-LLSVR

FIGURE 4: The final image restoration effect.

images in the training set and have better image restoration than the LR algorithm. The experimental results show that the combination of TFOA and LSSVR can achieve a satisfactory recovery effect. This method has advantages in other areas of competition, especially in terms of CPU time cost.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by National Key R&D Program of China (2017YFC0820700); the Foundation of Science and Technology on Information Assurance Laboratory (no. KJ-17-101); and the National Science Foundation Project of China (no. 61701188).



(a) Motion blurred images



(b) Disk blurred images



(d) Restored by LR









(e) Restored by the method proposed in this paper

FIGURE 5: The specific results.

References

- [1] M. R. Banham and A. K. Katsaggelos, "Digital image restoration," IEEE Signal Processing Magazine, vol. 14, no. 2, pp. 24-41, 1997.
- [2] J. G. Triana, L. C. Romero, and G. D. Roldán, "Restauración digital de imágenes mediante ecuaciones diferenciales parciales," Revista UDCA Actualidad & Divulgación Científica, vol. 16, no. 2, pp. 511-518, 2013.
- [3] A. Kumar, M. F. Hassan, and P. Raveendran, "Learning based restoration of Gaussian blurred images using weighted geometric moments and cascaded digital filters," Applied Soft Computing, vol. 63, pp. 124-138, 2018.
- [4] G. Xu and E. Tang, "Blind deconvolution for image restoration based on text characteristic," in Proceedings of the 2011 International Conference on Control, Automation and Systems Engineering (CASE), pp. 1-3, IEEE, Singapore, Singapore, July 2011.

- [5] J. Yao, S. Tian, X. Wang et al., "Blind restoration method of three-dimensional microscope image based on RL algorithm," in *Proceedings of the ISPDI 2013 - Fifth International Symposium* on Photoelectronic Detection and Imaging 2013: Optical Storage and Display Technology, vol. 8913, p. 89130Z, International Society for Optics and Photonics, Beijing, China, 2013.
- [6] J. Y. Zhao, Y. H. Wu, J. L. Jia et al., "Image restoration based on real time wave-front information," *Guangxue Jingmi Gongcheng* (*Optics and Precision Engineering*), vol. 20, no. 6, pp. 1350–1356, 2012.
- [7] S. S. Mukherjee, R. Chowdhury, and S. Bhattacharyya, "Image restoration using a multilayered quantum backpropagation neural network," in *Proceedings of the 2011 International Conference on Computational Intelligence and Communication Networks, CICN 2011*, pp. 426–430, IEEE, India, October 2011.
- [8] P. K. Sethy, L. Panda, and S. K. Behera, "Ann based image restoration in approach of multilayer perceptron," in *Proceedings of the 2016 International Conference on Inventive Computation Technologies, ICICT 2016*, vol. 2, pp. 1–4, IEEE, Coimbatore, India, August 2016.
- [9] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 349–366, 2007.
- [10] Y. Liu, C. Li, and H. Xu, "A BP neural network model for image restoration based on area around pixel and edge information," in *Proceedings of the 2nd Annual Conference on Electrical and Control Engineering, ICECE 2011*, pp. 3761–3764, IEEE, China, September 2011.
- [11] G. Sharma, F. Zhou, J. Liu et al., "Fruit fly optimization based least square support vector regression for blind image restoration," in *Proceedings of the International Symposium on Optoelectronic Technology and Application 2014: Image Processing and Pattern Recognition*, pp. 9301–93011W, International Society for Optics and Photonics, Beijing, China, 2014.
- [12] W.-T. Pan, "A new fruit fly optimization algorithm: taking the financial distress model as an example," *Knowledge-Based Systems*, vol. 26, pp. 69–74, 2012.
- [13] X. Guo, J. Zhang, W. Li, and Y. Zhang, "A fruit fly optimization algorithm with a traction mechanism and its applications," *International Journal of Distributed Sensor Networks*, vol. 13, no. 11, Article ID 1550147717739831, 2017.
- [14] F. Ye, X. Y. Lou, L. F. Sun, and W. Du, "An improved chaotic fruit fly optimization based on a mutation strategy for simultaneous feature selection and parameter optimization for SVM and its applications," *PLoS ONE*, vol. 12, no. 4, p. e0173516, 2017.
- [15] D. Li, R. M. Mersereau, and S. Simske, "Blind image deconvolution through support vector regression," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 18, no. 3, pp. 931– 935, 2007.
- [16] Z.-B. Jiang and Q. Yang, "A discrete fruit fly optimization algorithm for the traveling salesman problem," *PLoS ONE*, vol. 11, no. 11, Article ID e0165804, 2016.
- [17] Z. He, Q. Wang, Y. Shen, and Y. Wang, "Discrete multivariate gray model based boundary extension for bi-dimensional empirical mode decomposition," *Signal Processing*, vol. 93, no. 1, pp. 124–138, 2013.
- [18] X. Guo, Y. Li, T. Suo, H. Liu, and C. Zhang, "Dynamic deformation image de-blurring and image processing for digital imaging correlation measurement," *Optics and Lasers in Engineering*, vol. 98, pp. 23–30, 2017.

