Image Interpolation Scheme based on SVM and Improved PSO

X F Jia¹, B T Zhao¹⁺, X X Liu¹ and H P Song¹

¹School of Electrical and Information Engineering, Anhui University of Science & Technology, Huainan 232001, China
⁺zhaobaiting@gmail.com

Abstract: In order to obtain visually pleasing images, a support vector machines (SVM) based interpolation scheme is proposed, in which the improved particle swarm optimization is applied to support vector machine parameters optimization. Training samples are constructed by the pixels around the pixel to be interpolated. Then the support vector machine with optimal parameters is trained using training samples. After the training, we can get the interpolation model, which can be employed to estimate the unknown pixel. Experimental result show that the interpolated images get improvement PNSR compared with traditional interpolation methods, which is agrees with the subjective quality.

1. Introduction

Image interpolation is designed to reproduce a high-resolution image with a low-resolution image, which is a key technology in digital image and video processing. Image interpolation techniques are divided into traditional interpolation, edge-based interpolation and region-based interpolation. Traditional interpolation is simple to implement, but the image edge is blurred. Edge-based interpolation has a certain enhancement on the edge of the image, but requires an accurate extraction of the edges of the image. Region-based interpolation requires accurate segmentation of the region, and different regions need to design different interpolation formula. In recent years, various learning-based image interpolation methods have been proposed, such as neural network image interpolation [1], SVM image interpolation [2-3].

The choice of parameters is an important problem in the field of SVM research, and its essence is an optimization search process. Phan AV [4] et al designed a novel hybrid model to enhance SVM’s classification efficiency by using genetic algorithms. A SVM classifier [5] based on particle swarm optimization was proposed to increase the accuracy of classify electrical faults in radial distribution systems. Ji [6] et al introduced an iterative optimization algorithm applying the EnKF method to optimize SVM’s parameters.

SVM is a hot research topic in the field of machine learning and data mining. The research of parameter selection is not uncommon, but few studies on the background of image interpolation. In this paper, we improve the particle swarm optimization algorithm to optimize the parameters of the SVM, and then use the SVM to optimize the image interpolation. A high precision image interpolation method is proposed. The design flow is shown in Figure 1.
Figure 1. The flow chart of image interpolation

2. Theory

2.1. Support vector regression machine

The basic idea of the vector regression algorithm is to map the data $x$ to the high-dimensional feature space through a nonlinear mapping $\phi$ and linear regression in the space. The function regression problem can be described as: given a set of training samples $T = \{(x_i, y_i)\}_{i=1}^n$, construct a linear regression function in the feature space:

$$f(x) = \omega^T \phi(x) + b$$

Which, $\omega$ and $b$ represents the regression factor, is obtained by solving the minimization of the regularization risk function, i.e. [7]:

$$\min \frac{1}{2} \| \omega \|^2 + C \sum_{i=1}^{n} \left[ y_i - \omega \cdot \phi(x_i) - b \right]^2$$

s.t. $\omega \cdot \phi(x_i) + b - y_i \leq \epsilon + \xi_i^+$

$$\epsilon, \xi_i^+ \geq 0 \quad i = 1, \ldots, n$$

Which, $\| \cdot \|$ on behalf of the regularization operator, $C$ as a penalty factor and $\gamma$ for the insensitive parameter. Solving the formula (1), the final regression function can be obtained:

$$f(x) = \sum_{i=1}^{n} (\alpha_i - \alpha_i^+) \phi(x_i) \cdot \phi(x) + b = \sum_{i=1}^{n} (\alpha_i - \alpha_i^+) k(x_i, x) + b$$

Among them, $\alpha_i$ and $\alpha_i^+$ is the Lagrangian multiplier. $k(x_i, x) = \phi(x_i) \cdot \phi(x)$ as a kernel function, used to achieve non-linear mapping.

2.2. Particle swarm optimization

Particle swarm optimization (PSO) [8] algorithm is a new branch of group intelligence, which is an efficient stochastic optimization technique. In solving the PSO optimization problem, the solution lies in the search for the location of a bird in space, saying that these birds are "particles". Each particle has its own position and velocity and individual fitness corresponding to the objective function. According
to their own and group flight experience to adjust their flight path, to move closer to the best. Each adjustment of the flight trajectory is achieved by tracking two "extreme values", one is the optimal solution of the particle itself, called the individual extreme value \( p_i = (p_{i1}, p_{i2}, \ldots, p_{in}) \); the other is the optimal solution of the whole group, called the global extreme value \( p_g = (p_{g1}, p_{g2}, \ldots, p_{gn}) \). After the particles find the two optimal values, update their speed and position according to the formula (4-5):

\[
\begin{align*}
    v_{id}(t+1) &= w v_{id}(t) + c_1 \text{rand}(t)(p_{id}(t) - x_{id}(t)) + c_2 \text{Rand}(t)(p_{gd}(t) - x_{id}(t)) \\
    x_{id}(t+1) &= x_{id}(t) + v_{id}(t+1)
\end{align*}
\]

Where \( w \) is the inertia weight, \( d = 1, 2, \ldots, n \), \( t \) is the number of current evolutionary iterations, \( i = 1, 2, \ldots, m \), \( m \) is the population size, \( \text{rand}(t) \) and \( \text{Rand}(t) \) are the random number distributed between [0,1]; \( c_1 \) and \( c_2 \) are the two positive numbers, called the acceleration factor; \( v_{id}(t) \) is the first \( d \) dimension component of the flight velocity in the first \( t \) iteration of the first \( i \) particle; \( x_{id}(t) \) is the first \( d \) dimension component of the position of the first \( t \) iteration of the first \( d \) particle; \( P_{gd}(t) \) is the first \( d \) dimension component of the best position of the particle \( i \); \( P_{id}(t) \) the first \( d \) dimension component of the best position of the population.

3. SVM optimization parameters based on PSO

\( C \) is used to balance the training error and the model complexity. The model complexity of SVM increases as \( C \) increases, but, the training error decreases. When \( C \) increases to a certain extent, the performance of SVM will not change any more. When \( C \) small, the punishment of empirical error is little, low complexity training, poor accuracy of fitting. The \( \gamma \) reflects the sensitivity of the regression model to the noise contained in the input variable, The greater the insensitive parameter, the lower the accuracy of the model fitting, the lower the complexity, prone to over-fitting. The kernel function \( \sigma \) represents the mean square error of the Gaussian function. When \( \sigma \) is small, the fitting function of the RBF kernel function is better, but the generalization ability is deteriorated.

3.1. Parameter optimization process

In this paper, we discuss the optimization of parameters \( C, \gamma \) and \( \sigma \) in the case of use \( \gamma \)-SVR and the RBF kernel function.

The PSO optimization algorithm is as follows:

\( \text{① Initialization. Initialize the acceleration factor } c_1, c_2 \text{ and the inertia weight } w \text{ and the maximum number of evolutionary iterations } t \text{ to determine the population size. The parameters to be optimized are mapped to particle swarm, and the initial position and velocity of each particle are obtained in a random manner;} \)

\( \text{② Calculate the fitness value. The fitness value of each particle is calculated according to the fitness function;} \)

\( \text{③ Determine the individual extremes and global extremes;} \)

\( \text{④ Update the speed and location of each particle. Using the formula (4), (5) to generate a new generation of population;} \)

\( \text{⑤ Update individual extremes and global extremes. Calculating the fitness value of each particle in the new particle group according to the fitness function, and then updating the individual extreme and the global extreme value according to the fitness value;} \)

\( \text{⑥ If } f(p_i) > P_{id}, \text{ update the individual extreme } , \text{ if } f(p_g) > P_{gd}, \text{ update the global extreme } ; \)

\( \text{⑦ Cross operation. The particles } m \times p \text{ are randomly selected according to the preset probability } p_x, \text{ For each pair of particles were bit by bit "take large" operation and "take small" operation to produce two new particles;} \)
⑧ Calculate the fitness values for new particles. If the fitness value is better than the fitness value of the corresponding particle before the cross operation, accept them and update the best position of the individual and the group; otherwise return to the state before crossing;

⑨ Check the termination condition. When the required evolutionary algebra or the maximum adaptation of the 15th generation did not change, terminate the operation. The particles corresponding to the global optimal position are mapped into parameters $C$, $\gamma$ and $\sigma$ otherwise go to step 4.

### 3.2. Fitness function

The fitness function is the only deterministic indicator of the merits of the particle position. Select the function that reflects the SVR regression performance as a fitness function, defined as:

$$ F(x_i) = 1 / \left( \frac{1}{n} \sum_{i=1}^{n} | y_i - \hat{y}_i | \right) $$

$F(x_i)$ is the fitness value of the first $i$ particle in the population, $y_i$ and $\hat{y}_i$ are expected output value and training output value corresponding to the SVR corresponding to the parameter in the first $i$ particle respectively.

### 4. SVR interpolation algorithm

The advantages and disadvantages of image interpolation are closely related to the correlation of images. Correlation refers to the same local area of the adjacent pixels of an image have similar gray values. The gray value of the image to be interpolated is closely related to the average gray value of the pixel in the neighborhood and the change in the gray value of each direction pixel [3]. Therefore, SVR training samples should include them.

![Figure 2. Interpolation mode](image)

The interpolation process is illustrated by taking the image enlarged in 2 times in both the row direction and the column direction. The input of the training sample and the test sample are constructed as 22 dimensions, the former output is the known pixel value, and the latter output is the pixel value to be interpolated. The 22-dimensional inputs are the known pixel values $P_i, i = 1, 2, \ldots, 6$ contained in Figure 2, the mean $\text{avg}$ of the six known pixels, and the difference between the two pixels in the horizontal, vertical, and diagonal directions $d_m, m = 1, 2, \ldots, 15$:

$$ \text{avg} = \frac{1}{6} \sum_{i=1}^{6} P_i, \quad d_1 = P_1 - P_2, \quad d_2 = P_1 - P_3, \quad d_3 = P_1 - P_4, \quad d_4 = P_1 - P_5, \quad d_5 = P_1 - P_6 $$

$$ d_6 = P_2 - P_3, \quad d_7 = P_2 - P_4, \quad d_8 = P_2 - P_5, \quad d_9 = P_2 - P_6, \quad d_{10} = P_3 - P_4, \quad d_{11} = P_3 - P_5, \quad d_{12} = P_3 - P_6 $$

$$ d_{13} = P_4 - P_5, \quad d_{14} = P_4 - P_6, \quad d_{15} = P_5 - P_6 $$

Using all the training samples to train the SVR to get the interpolation model, and then use the test sample as the input of the interpolation model to estimate the gray value to be interpolated. The interpolation is done in the following three steps:
① Interpolate even rows odd columns: according to Figure 2(a) interpolation model construction training samples;
② Interpolate even rows and columns: according to Figure 2(b) interpolation model construction training samples, when constructed, $p_r$ and $p_c$ are zero;
③ Interpolate odd rows even columns: according to Figure 2(c) interpolation model construction training samples.

5. Experimental results
The experiments were performed by LIBSVM [9] and MatlabR2009b. First, 1/2 under-sampling the standard image, then, interpolating the obtained image with different methods, such as linear interpolation, cubic cubic interpolation, nn interpolation, SVM interpolation and PSO optimization interpolation. In the simulation process, real-valued codes are used to randomly generate 3-dimensional particles as the initial positions of the particle swarm. Randomly generating random numbers in [0, 1] as the initial velocity of the particles. In order to solve the problem of position and velocity update of different parameters, the three parameters are normalized before optimization, and then reverse the results of the optimization. The range of the search parameters is $C \in (0.001, 150)$, $\gamma \in (0, 0.99)$, $\sigma \in (0.001, 2)$. PSO population size $M=50$, maximum evolutionary algebra $T=100$, crossover probability $P_c=0.4$, inertia weight $w=1$, acceleration factor $c_1= c_2=2$. The global optimal particles obtained after optimization were (31.6208, 0.9204, 0.0714).

From table 1 we can find that the proposed interpolation effect is better than other methods, the PSNR values are higher than those of linear interpolation, cubic interpolation, nn interpolation and SVM interpolation. Figures 3 show the interpolation results of Lena in different methods. It can be seen that the result of linear interpolation, cubic interpolation and nn interpolation in Lena's hair, hat edge appear obvious saw-tooth. nn interpolation has an edge blur on the right side of Lena and the right side of the hair. SVM interpolation effect than the previous three have significantly improved, but in the shoulder of Lena there are obvious jagged effect, as shown in figure 4, it is the effect of enlarging the residual image of the shoulder in the white frame in Lena standard image in figure 3 by 350 times.

|                | Linear interpolation | Cubic interpolation | nn interpolation | SVM interpolation | PSO optimized interpolation |
|----------------|----------------------|---------------------|------------------|-------------------|-----------------------------|
|                | PSNR | NMSE | PSNR | NMSE | PSNR | NMSE | PSNR | NMSE | PSNR | NMSE | PSNR | NMSE |
| Lena           | 23.26 | 0.0175 | 22.86 | 0.0192 | 24.05 | 0.0152 | 29.96 | 0.0037 | 30.31 | 0.0035 |
| Cameraman      | 21.42 | 0.0265 | 20.72 | 0.0307 | 21.51 | 0.0263 | 28.03 | 0.0057 | 28.42 | 0.0056 |
| Peppers        | 23.63 | 0.0158 | 23.28 | 0.0172 | 25.20 | 0.0110 | 30.34 | 0.0034 | 30.69 | 0.0033 |
| Boattest       | 21.49 | 0.0242 | 20.73 | 0.0289 | 21.90 | 0.0220 | 27.69 | 0.0058 | 27.93 | 0.0056 |
| Clock          | 22.22 | 0.0103 | 21.72 | 0.0116 | 23.20 | 0.0082 | 31.14 | 0.0013 | 31.55 | 0.0012 |
| House          | 25.30 | 0.0090 | 25.07 | 0.0095 | 24.12 | 0.0119 | 31.94 | 0.0020 | 32.47 | 0.0018 |
| Average        | 22.89 | 0.0172 | 22.40 | 0.0195 | 23.33 | 0.0158 | 29.85 | 0.0037 | 30.23 | 0.0035 |

| Standard image | Linear interpolation | Cubic cubic interpolation | nn interpolation | SVM interpolation | PSO optimized interpolation |
|----------------|----------------------|-----------------------------|------------------|-------------------|-----------------------------|

Figure 3. Lena interpolation results of each method
6. Conclusion
In the PSO algorithm, the crossover operator based on the principle of "taking large" and "taking small" can solve the problem of updated the new speed and the new position beyond the boundary. The improved SVM parameters can be obtained by optimizing the PSO parameters. The SVM with optimal parameters can be used to obtain interpolation of images with significant improvement of subjective and objective indexes in the case of full consideration of image correlation. It can effectively prevent the presence of zipper effects in the image, and reduces edge blurring.

Acknowledgement
This research was financially supported by the following funds: The National Natural Science Foundation of China (61501006), the Postdoctoral Fund National Projects (2016M592035) and Anhui Provincial Natural Science Research Projects in Central Universities (KJ2017A076).

References
[1] Plaziac N 1999 IEEE Transactions on Image Processing vol 8 p 1647  
[2] X F Jia, B T Zhao, M R Zhou and W Gao 2015 Application Research of Computers vol 32 p 623  
[3] L Y Ma, Y Shen and J C Ma 2008 Journal of Systems Engineering and Electronics vol 19 p 618  
[4] Phan AV, Nguyen ML and Bui 2017 Applied Intelligence vol 46 p 455  
[5] Ming-Yuan C and Thi T H 2017 Computational Intelligence and Neuroscience p 1  
[6] Y S Ji, Y S Chen, H H Fu and G W Yang 2017 Pattern Recognition vol 62 p 202  
[7] Vapnik V 1995 Springer  
[8] J Kennedy R Eberhart 1995 IEEE International Conference on Neural Networks p 1942  
[9] C C Chang and C J Lin 2001 Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm