An Image Optimization Model Based on NAG-Enhanced Inexact Block Coordinate Descent and Super-Resolution Reconstruction

Huihong Zheng
Department of Food, MinBei Vocational and Technical College, Nanping, 353000, China
E-mail: huihonghhzh@163.com

Keywords: IBCD, NAG, SR, mean-shift, image processing

Received: August 25, 2025

Image processing technology is often combined with deep learning to be applied in various recognition and classification tasks. However, most existing image processing technologies struggle to balance both accuracy and efficiency. To address this issue, this paper proposes an efficient image processing model that embeds Nesterov's accelerated gradient optimization imprecise block coordinate descent algorithm into a super-resolution reconstruction framework. The model first divides the input image into blocks and uses the Mean Shift algorithm to cluster the image blocks to reduce computational complexity. Subsequently, the Nesterov accelerated gradient algorithm was used to accelerate the descent process of the imprecise block coordinate algorithm, and finally the image resolution was improved through super-resolution reconstruction. The experiment is based on the DIV2K dataset, with the alternating direction method of multipliers, stochastic gradient descent, and Super-Resolution Convolutional Neural Network models as comparison baselines, and it is validated in a hardware environment equipped with an AMD Radeon RX 6800 XT GPU. The results show that the average Peak Signal to Noise Ratio and Structural Similarity Index of the proposed algorithm are 33.43dB and 0.916, respectively. The average processing time for a single image is 16ms, and the overall similarity of the images is 90.1%. The minimum image processing time for the proposed image processing model is 36ms, and the processing time for a single image does not exceed 40ms. These results demonstrate that the proposed model not only ensures high accuracy but also meets the efficiency requirements, allowing for effective restoration and optimization of target images. The proposed method offers new insights for image processing and contributes to the optimization of various techniques based on image processing.

Povzetek: Predstavljen je model za optimizacijo slik, ki združuje pospešeni gradient NAG, netočni blokovni koordinatni spust in superločljivost. Rezultati na podatkovni zbirki DIV2K kažejo visoko kakovost rekonstrukcije slik ter zelo kratke čase obdelave, kar potrjuje ravnotežje med točnostjo in učinkovitostjo.

1 Introduction

With the development of deep learning, various practical technologies based on image recognition have emerged. Intelligent algorithms and devices can extract and analyze large amounts of feature information from images to complete simple tasks such as object recognition and classification [1]. Compared to manual methods, these technologies offer advantages in terms of efficiency and durability. However, these techniques heavily rely on image quality, which directly affects the accuracy of the final results. While numerous image processing optimization methods have been proposed by researchers to address this issue, current methods often fail to balance accuracy and efficiency. Therefore, there is an urgent need for an image optimization technology that ensures both high efficiency and accuracy for tasks like intelligent classification and recognition based on image recognition. The Inexact Block Coordinate Descent (IBCD) algorithm reduces computational complexity by performing non-precise iterative block updates, which provides significant efficiency when processing large-scale images, but it often faces convergence issues that lead to poor results [2]. On the other hand, the Nesterov Accelerated Gradient (NAG) algorithm accelerates convergence through foresight gradient updates while maintaining the accuracy of convergence [3]. To solve the efficiency problem in image optimization, this study proposes an efficient image processing algorithm based on NAG-optimized IBCD. The Super-Resolution (SR) algorithm aims to improve image quality by reconstructing low-resolution images into high-resolution ones through denoising and detail enhancement. Meanwhile, the Mean Shift Clustering (Mean-Shift) algorithm reduces computational complexity by clustering data and enhances accuracy by reducing bias during parameter changes [4-5]. To address the low accuracy of image processing technologies, this study proposes an image extraction method based on SR and Mean Shift algorithms. Finally, by combining these two methods, the study presents an image processing model that optimizes IBCD and SR. This model innovatively balances efficiency and image quality by embedding the NAG-optimized IBCD into SR. The

model enhances image quality through optimization based on resolution. It is expected that this model can solve the problem of balancing processing efficiency and accuracy in the field of image processing, and promote the development of other technologies based on image recognition. The core objective of this study is to verify that the NAG-optimized IBCD algorithm outperforms traditional optimization methods in terms of convergence speed, and to demonstrate that the model that integrates the improved algorithm with SR and Mean Shift can significantly improve image quality, in order to break through the performance bottleneck of traditional image processing methods in image detail restoration and structure preservation.

2 Related works

IBCD has been widely used by scholars at home and abroad in the study of various problems due to its high computational efficiency and ability to process large amounts of data simultaneously. For example, in order to minimize the approximate value of the objective function, Birgin and Martínez proposed a general algorithm for minimizing quadratic models with quadratic regularization on variable blocks based on IBCD, and proved the convergence and complexity of the algorithm. The effectiveness of the algorithm was verified in the solution of the continuous version of the traveling salesman problem [6]. In order to optimize non-smooth blocks in non-convex environments, Rebegoldi proposed an optimization method based on IBCD. This method ensured that the inexact proximal operator converged to a stationary point of the objective function by solving each block subproblem in a finite number of steps. The results showed that this method could be effectively applied to the solution of block coordinate problems [7]. In order to more quickly obtain the optimal solution of non-convex, convex, and strongly convex objective functions, Farias and Gutman proposed a convergence analysis method based on IBCD. This method improved the convergence speed by reducing the expected gradient norm and the suboptimality gap. The results showed that the convergence rate of this method for the optimal solution was relatively efficient [8]. In order to successfully separate the components in a complex chemical system, Wang et al. proposed a fast calculation method for generalized scattered block optimization problems based on IBCD. The method used the finite internal iteration of the alternating proximal gradient method to obtain an inexact solution so that the iterative sequence converges to a stable point. The results showed that the method was effective [9]. In order to maximize the daily throughput between drones and base stations, Van et al. proposed an optimization algorithm based on IBCD to solve mixed-integer nonlinear programming problems. The algorithm uses the exact penalty of mathematical programming under equilibrium constraints to relax integer constraints. The results showed that the method was more effective than several benchmark methods [10].

The development of image processing technology has also produced some relatively mature optimization theories and practical technologies, and many scholars have conducted in-depth research in this field. For example, in order to improve the recognition accuracy of artificial intelligence algorithms, Wang et al. proposed an image optimization scheme for machine vision. This scheme combined dedicated variable bit rate coding on the basis of end-to-end compression to improve image quality. In the test of two different data sets, the image bit rate was increased by 31.69% and 23.96% respectively [11]. In order to improve the accuracy of the algorithm in identifying human skin diseases, Hassan et al. proposed multiple image optimization algorithms based on strategies such as stochastic gradient descent, random momentum optimization descent, and deep integration. The final results showed that the use of optimizers such as SGD and Adam could effectively improve the accuracy of image recognition [12]. In order to reduce the impact of haze weather on intelligent transportation systems, Sahu et al. proposed a defogging method that cascaded the brightness preservation model with the color channel based on the parameter-adaptive dual-channel improved simplified pulse-coupled neural network. This method could effectively optimize the foggy images received by smart devices [13]. Frants et al. aimed to maximize the optimization effect of image dehazing and proposed an encoder-decoder architecture based on a robust quaternion neural network. The architecture used quaternion data flow, quaternion pixel loss function, and quaternion instance normalization layer to improve the optimization effect. The results showed that this method improved the accuracy and recall rate in the image dehazing optimization process [14]. In order to solve the problem of residual artifacts in the low-light image conversion process, Wang et al. proposed a one-to-many relationship normalization model. The model mapped the low-light image to a Gaussian distribution reversible network and constrained the image loss function by modeling the conditional distribution of the image. Experimental results showed that this method could effectively reduce image artifacts [15]. The summary table of the above-mentioned related work is shown in Table 1.

Table 1: Summary table of related work

Literature	Method	Performance metrics	Contribution	Limitation
Birgin et al. [6]	A general algorithm for minimizing the quadratic model of variable block quadratic regularization based on IBCD	Convergence analysis, computational complexity	Suitable for a wide range of smooth non convex problems	Slow convergence of high-dimensional problems
Rebegoldi [7]	Variable scale block coordinate method with proximal error	Converge to the stationary point	Capable of handling non convex and non smooth problems	The convergence speed is not clearly quantified
Farias et al. [8]	Random Block Coordinate Descent Method under Hölder Smooth Setting	Expected gradient norm, convergence rate	Obtained a faster convergence rate	Hölder constant may be difficult to determine
Wang et al. [9]	IBCD; Alternating proximal gradient method	Shows faster convergence and lower error on both synthetic and real data	Efficient processing of tensor decomposition with constraints	Universality to be verified
Van et al. [10]	Mixed integer nonlinear programming algorithm based on Block Coordinate Descent	Daily throughput increased by 19.7%	Effectively solve UAV trajectory optimization	High algorithm complexity
Wang et al. [11]	End to end image compression framework for machine vision	Image bitrate increased by 31.69% and 23.96%	Coordinated optimization of compression and recognition tasks	The subjective quality of reconstructed images may not be optimal
Hassan et al. [12]	Comparative study of multiple optimizers	The classification accuracy using Adam optimizer reaches 96.18%	Comprehensively compared the performance of different optimizers	No new optimization methods have been proposed
Sahu et al. [13]	Simplified pulse coupled neural network based on parameter adaptive dual channel improvement	Obtained Peak Signal to Noise Ratio of 28.6 dB	Single image dehazing is effective, with adaptive parameters	High computational cost
Frants et al. [14]	Dehazing based on quaternion neural network	Obtain Peak Signal to Noise Ratio of 32.41 dB on SOTS indoor dataset	Good color fidelity, comprehensive processing of color information	The computational complexity is higher than that of real valued networks
Wang et al. [15]	Low light image enhancement based on standardized flow	Obtain Peak Signal to Noise Ratio of 24.3dB on the LOL dataset	Learnable complex distributions	Requires a larger model and computational complexity

In summary, existing image processing methods have made some progress in theory and practical applications, but the current image processing methods are either low in accuracy or take a long time to process. In order to solve the limitations of existing image processing methods, this study proposes a fast image optimization algorithm based on NAG improved IBCD and a high-precision image extraction algorithm based on pixel reconstruction by SR. An efficient image processing model is constructed based on the two algorithms, and it is expected that the model can achieve the fusion of high efficiency and high precision in the image processing process and be promoted in practical applications.

3 Construction of image processing model based on IBCD and SR algorithm

3.1 Design of optimization algorithm based on IBCD and NAG

Image processing is a key step in all technologies that make decisions and identify objects based on image feature information. The quality of the image obtained in this process cannot be ignored. At the same time, information judgment based on images is often used in fields with large amounts of information and high timeliness requirements, such as real-time detection and data classification [16]. Therefore, in the image

processing process, processing speed and image quality are important indicators for evaluating processing performance. The core mechanism of the IBCD algorithm is to decompose complex high-dimensional problems into multiple low-dimensional modules through dimensional decoupling, and use the variable alternating fixed strategy to centrally optimize the target module to solve simple mathematical problems in the low-dimensional module [17]. This method can decompose images rich in two-dimensional information into simple block coordinates, significantly improving the processing speed. The specific process of image processing using IBCD is shown in Figure 1.

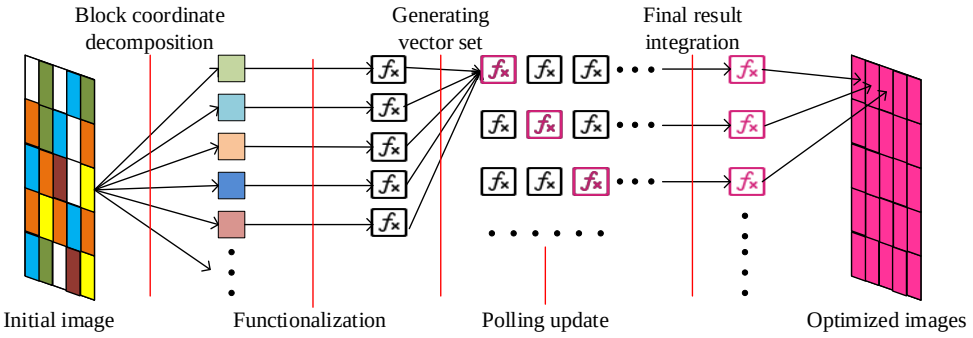


Figure 1: Image processing flow of the IBCD algorithm

In Figure 1, before processing the image, the image is first divided into blocks, transforming the complex multidimensional function problem into simple one-dimensional mathematical function problems for each block. A certain type of data in the image is regarded as the optimization index and converted into an abstract energy minimization function, which typically consists of a data fidelity term and a regularization term, as shown in Equation (1).

$$F(u)=H(u)+\gamma R(u) \quad (1)$$

In Equation (1), u represents the image data that needs to be optimized. $H(u)$ represents the data fidelity term, which is used to measure the degree of agreement between the solution u and the observed data. It can usually be expressed as $H(u)=\frac{1}{2}\|u-x_r\|^2$, x_r represents the true data value corresponding to the image point. $R(u)$ represents the regularization term, used to introduce prior constraints, usually represented as $R(u)=\gamma\|\nabla u\|$, ∇u represents the image gradient. γ represents the regularization parameter. Then, the data is updated by block polling, with the calculation Equation shown in Equation (2).

$$\begin{cases} x_1^k=\arg\min g(x_1,x_2^{k-1},x_3^{k-1},...,x_n^{k-1}) \\ x_2^k=\arg\min g(x_1^k,x_2,x_3^{k-1},...,x_n^{k-1}) \\ x_3^k=\arg\min g(x_1^k,x_2^k,x_3,...,x_n^{k-1}) \\ ..... \\ x_n^k=\arg\min g(x_1^k,x_2^k,x_3^k,...,x_n) \end{cases} \quad (2)$$

In Equation (2), x represents the vector at the function problem decomposition point, with a count of n, $\arg\min g$ denotes the search for the minimum value, and k represents the number of iterations. The function values within the blocks are repeatedly updated until the termination condition or the maximum iteration count is reached. The termination condition in this process is shown in Equation (3).

$$\left\|E_n^{k+1}-E_n^k\right\|\le\eta \quad (3)$$

In Equation (3), η represents the convergence threshold. To ensure the image processing rate, fewer iterations are typically used in block coordinate descent, which may result in many functions within the blocks not reaching the optimal solution at the termination point. However, increasing the number of iterations can cause exponential growth in processing time. The NAG algorithm can optimize the momentum direction by predicting the gradient change trend and combine dynamic adjustment of the update step size to suppress iteration oscillations, allowing the objective function to converge in fewer iterations or with shorter iteration steps [18]. The specific process of the NAG algorithm is shown in Figure 2.

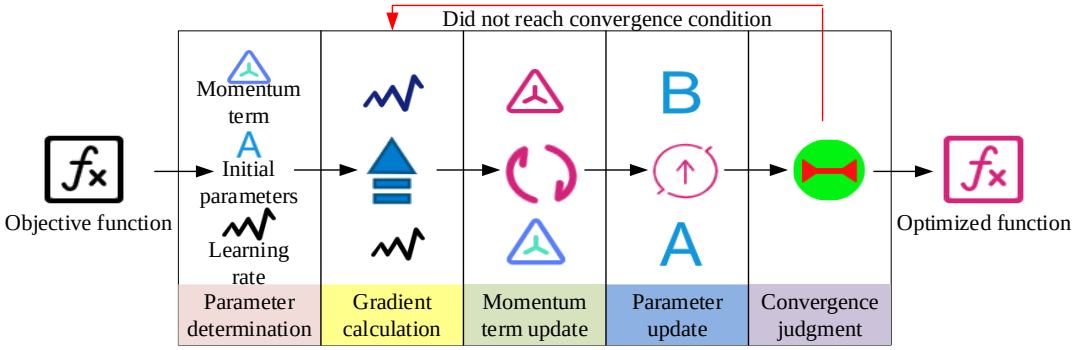


Figure 2: NAG algorithm operation flow chart

In Figure 2, when using the NAG algorithm to optimize another algorithm, it is necessary to first determine the algorithm's momentum, initial parameters, and learning rate, then perform a random iteration. Afterward, the gradient change and momentum direction are predicted according to the iteration step length, and the relationship between the changing gradient and function value is shown in Equation (4).

$$\begin{cases} y_{e+1} = z_e - s\Delta f(z_e) \\ z_{e+1} = y_{e+1} + \dfrac{e}{e+3}(y_{e+1} - y_e) \end{cases} \quad (4)$$

In Equation (4), y_{e+1} and z_{e+1} represent the main sequences of the e-th and $e+1$-th iterations, y_e and y_{e+1} represent the auxiliary sequences of the e-th and $e+1$-th iterations, $\Delta f(z_e)$ represents the gradient of the objective function at z_e, s is the iteration step length, and e represents the number of iterations. Then, the momentum direction in the function iteration process is updated, and the relationship between momentum direction and function value is shown in Equation (5).

$$f(z_e) - f(z^*) \geq \frac{3L\left\|z_0 - z^*\right\|^2}{32(e+1)^2} \quad (5)$$

In Equation (5), z^* represents the optimal solution of the function, z_0 is the initial value for the iteration, and L is the Lipschitz constant. After updating the momentum direction, other parameter values are updated, and convergence is checked for the updated parameters. When the function converges, the relationship between the number of iterations and the solution is shown in Equation (6).

$$f(z_e) - f(z^*) \leq O\left(\frac{M\left\|z_0 - z^*\right\|}{e^2}\right) \quad (6)$$

In Equation (6), O represents the asymptotic upper bound of the algorithm. If the above condition is satisfied, the optimized function is output. The study uses the NAG algorithm to optimize the IBCD algorithm, accelerating its convergence speed to improve the quality of the solutions of the IBCD function. Finally, the NAG-optimized IBCD image processing algorithm is obtained, which is named the NI algorithm. The specific process of this algorithm is shown in Figure 3.

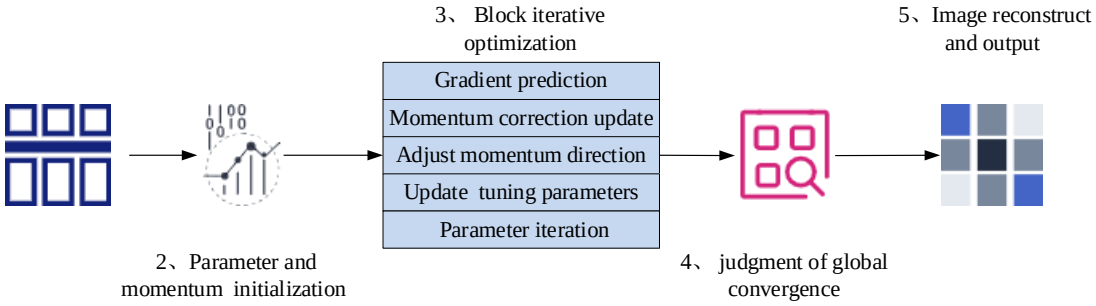


Figure 3: NI algorithm image processing flowchart

In Figure 3, the NI algorithm processes the image through five main stages: image block division and target definition, parameter initialization and momentum configuration, block iteration optimization, global convergence judgment, and image reconstruction output. First, the input image is divided into multiple regular small blocks, each modeled independently as a low-dimensional mathematical problem, simplifying the complex high-dimensional image processing task into block-wise optimization problems. Next, optimization parameters are set for each image block, and the momentum mechanism is introduced. Then, all image blocks are traversed in sequence, performing gradient prediction, parameter position estimation, momentum

correction updates, momentum direction adjustment, and parameter iteration during the optimization process. When the global judgment module detects that the overall energy function of all blocks is below the preset threshold or the maximum iteration count is reached, the optimization iteration stops. The optimized block parameters are then stitched according to the original spatial position to generate the final processed image.

3.2 Construction of image processing model based on SR and NI algorithms

The NI algorithm can only optimize existing image data. To achieve image processing, it must first be integrated with specific image information indicators, and then adjust specific parameters using the NI algorithm to achieve image optimization. The SR reconstruction algorithm used in this study is based on the classic Convolutional Neural Network (CNN) architecture, mainly consisting of a feature extraction layer, a nonlinear mapping layer, and an upsampling reconstruction layer. It is trained end-to-end by minimizing the Mean Squared Error (MSE) between the reconstructed image and the real high-resolution image. Among them, the feature extraction layer uses convolutional layers to preliminarily extract feature maps from low-resolution input images. The nonlinear mapping layer is composed of multiple residual blocks stacked together, used for deep learning and mapping of low-resolution features to high-resolution features. The upsampling reconstruction layer uses sub-pixel convolution layers to upsample and reconstruct the learned high-dimensional features into the final high-resolution image. SR technology extracts features from low-quality images and reconstructs them into high-quality images [19]. Therefore, the SR algorithm is used as the execution basis of the NI algorithm, with resolution serving as the specific indicator for controlling image quality during image processing. The specific process of the SR algorithm for image resolution extraction and reconstruction is shown in Figure 4.

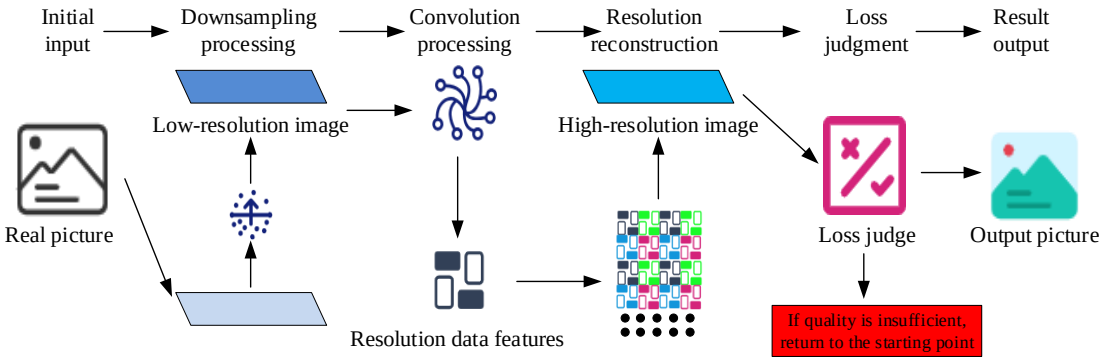


Figure 4: Image super-resolution reconstruction flowchart

In Figure 4, before image reconstruction, the real image is down-sampled into a low-resolution image. The calculation method for this process is shown in Equation (7).

$$P_l = D(\mathrm{P_r}; \theta_D) \quad (7)$$

In Equation (7), P_l and $\mathrm{P_r}$ represent the generated low-resolution image and the real image, respectively, while D and θ_D represent the down-sampling calculation process and related parameters. Then, the resolution features inside P_l need to be obtained through a feature extraction process, with the calculation Equation for this process shown in Equation (8).

$$P_f = F(P_l; \theta_f) \quad (8)$$

In Equation (8), P_f represents the extracted image features, and F and θ_f represent the feature extraction calculation process and related parameters. In the subsequent image reconstruction process, the extracted image resolution features must be mapped to the corresponding high-resolution space, as shown in the calculation process of Equation (9).

$$P_h = B(P_f; \theta_b) \quad (9)$$

In Equation (9), P_h represents the high-resolution image after mapping, and B and θ_b are the calculation process and related parameters for mapping. Finally, the mapping performance is evaluated using the loss function between the high-resolution image and the real image. If the effect is unsatisfactory, the relevant parameters in the mapping process are adjusted, and mapping is performed again. The calculation process for this is shown in Equation (10).

$$Loss = L(\mathrm{P_r}; P_h) \quad (10)$$

In Equation (10), L represents the specific evaluation index used in this process. The Mean-Shift algorithm can accurately cluster similar data groups without presetting the number of clusters by iterating through the density gradient to find data density peaks, making it suitable for image segmentation, object tracking, and other scenarios [20]. To further improve the efficiency and accuracy of image processing, the study uses the Mean-Shift algorithm to cluster the down-sampled low-resolution regions, reducing the number of block coordinate decompositions and the frequency of parameter

adjustments in the NI algorithm. The schematic diagram of the Mean-Shift algorithm is shown in Figure 5.

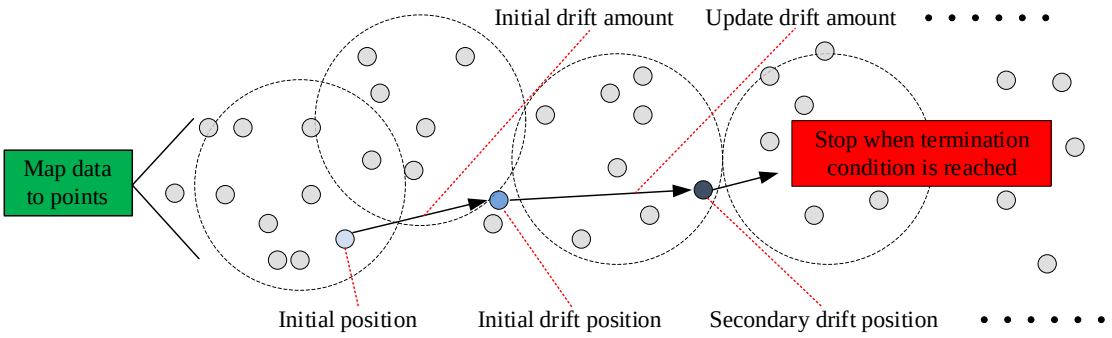


Figure 5: Mean-shift algorithm workflow schematic

In Figure 5, the first step is to assume that the data consists of initial points and drift space. At this point, the initial point distribution must satisfy a certain rule, as shown in Equation (11).

$$M(x)=\frac{1}{nh^{a}}\sum_{i=1}^{n}C(\frac{x-x_i}{h}) \quad (11)$$

In Equation (11), $M(x)$ represents the probability distribution function of the point set, n is the total number of points, h is the window length for traversal, C is the kernel function, a is the spatial dimension, and i is the point index. The mathematical relationship that the kernel function must satisfy is shown in Equation (12).

$$C(x)=\lambda(\|x\|^2) \quad (12)$$

In Equation (12), λ represents the normalization function. The next step is to compute the gradient change value of the probability distribution function, as shown in Equation (13).

$$\Delta M(x)=\frac{2\lambda}{nh^{a+2}}\sum_{i=1}^{n}(x_i-x)C'(\left\|\frac{x-x_i}{h}\right\|^2) \quad (13)$$

In Equation (13), C' represents the inverse of the kernel function C. Finally, the offset for clustering judgment is calculated, as shown in Equation (14).

$$m_h=\sum_{i=1}^{n}x_iC'(\left\|\frac{x-x_i}{h}\right\|^2)\div\sum_{i=1}^{n}C'(\left\|\frac{x-x_i}{h}\right\|^2) \quad (14)$$

In Equation (14), m_h represents the drift value, and each point is then adjusted by adding the drift value until the termination condition is reached. The termination judgment Equation for this process is shown in Equation (15).

$$\|m_h(x)-x\|\le\varepsilon \quad (15)$$

In Equation (15), ε represents the allowable error. The Mean-Shift algorithm-optimized SR algorithm serves as the execution framework for the NI algorithm, with resolution reconstruction as the optimization indicator. This results in the SMNI model, based on the SR algorithm and Mean-Shift algorithm for image processing. The process of this model is shown in Figure 6.

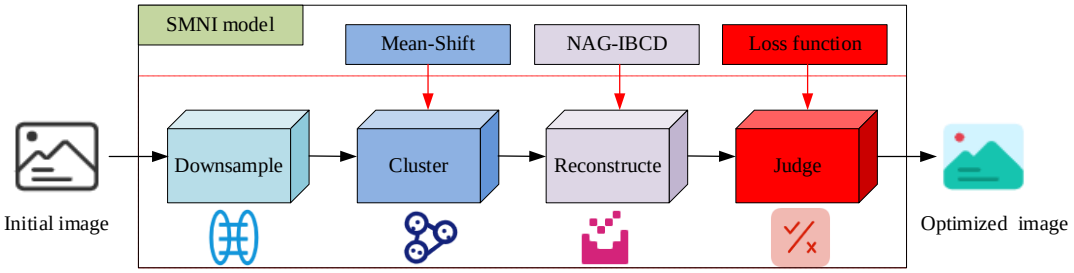


Figure 6: SMNI model workflow schematic

In Figure 6, the principle of the SMNI model is that the NI algorithm replaces the feature extraction and output structure originally present in the SR algorithm. During image processing, the real image is first converted into a low-resolution image via down-sampling. Then, the overall image is divided into multiple low-dimensional coordinate blocks, with the model incorporating the Mean-Shift algorithm to perform clustering analysis on the coordinate blocks. The coordinate blocks are clustered based on similar resolution features, and during the subsequent block coordinate descent optimization, no extra block-wise processing is required. Instead, the same

low-dimensional function optimal values for blocks within the same cluster are directly calculated based on the Mean-Shift algorithm clustering. The core of reducing the complexity of IBCD through this mechanism lies in transforming block-based optimization into cluster-based optimization. In traditional IBCD, each image block is treated as an independent variable block for iterative updates. In the SMNI model, after Mean Shift clustering, image blocks with similar resolution and texture features are grouped into the same cluster. The model no longer performs individual block coordinate descent updates for each block within the cluster, but instead calculates a representative shared solution for the entire cluster. Therefore, the IBCD algorithm reduces the number of variable blocks that need to be directly processed and updated from the "total number of blocks" to the "number of clusters", significantly reducing computational complexity and the frequency of parameter adjustments. Due to the effect of the NAG algorithm, the optimal solution for the low-dimensional function can be obtained with shorter steps and fewer iterations. Finally, when the required image quality or maximum iteration count is reached, the resolution information is integrated and output, resulting in a high-resolution, high-quality image.

4 Performance verification of image processing model based on IBCD

4.1 Performance verification of IBCD algorithm based on NAG optimization

To verify the practical performance of the proposed method, the NI algorithm was first tested. The test used the DIV2K dataset as the experimental sample. The DIV2K dataset contains 1000 2K images, including LR-HR image pairs generated by bicubic downsampling with unknown degradation, and supports iterative training and performance evaluation of image processing algorithms. The Alternating Direction Method of Multipliers (ADMM) algorithm, Support Vector Machine (SVM), and Stochastic Gradient Descent (SGD) algorithm were used as controls. During the experiment, all algorithms were carried out under the same laboratory conditions. The specific hardware and software equipment used in the experiment are shown in Table 2.

Table 2: Experimental equipment and environment

/	Category	Version	Category	Version
Software	Operating System	Windows 11	Deep Learning Framework	TensorFlow 2.5
	Programming Language	Python 3.9	Visualization Library	Seaborn
Hardware	CPU	Intel Core i9-11900K	GPU	AMD Radeon RX 6800 XT
	Memory	64GB RAM	Storage	512 GB NVMe SSD

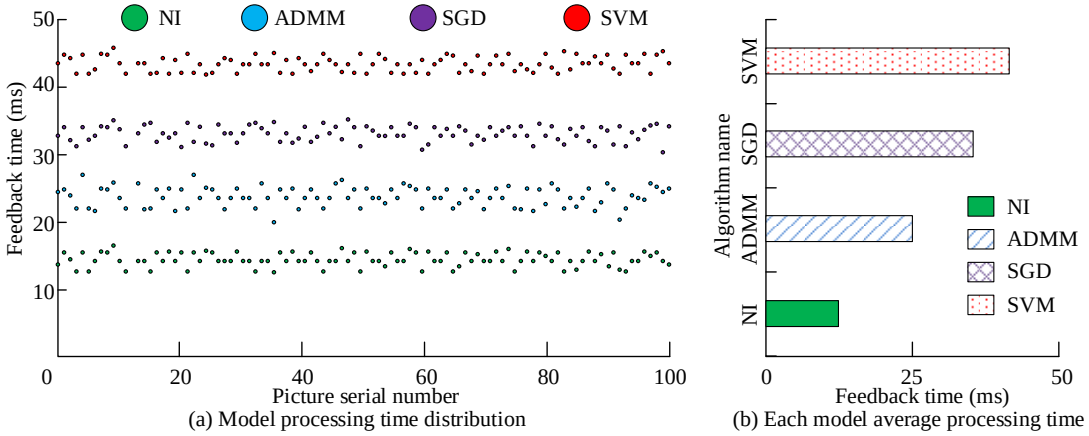
In order to verify the superiority of the NI algorithm, Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index (SSIM) were used as evaluation metrics to compare the proposed model with three advanced image super-resolution models: Super Solution Convolutional Neural Network (SRCNN), Enhanced Deep Super Solution Network (EDSR), and Residual Channel Attention Network (RCAN). Each algorithm was run independently 10 times to ensure result stability. The comparison results of PSNR and SSIM for the four models are shown in Table 3.

Table 3: Comparison of PSNR and SSIM for Four Models

Models	PSNR/dB	SSIM
SRCNN	30.24 ± 0.35	0.871 ± 0.008
EDSR	32.76 ± 0.28	0.894 ± 0.006
RCAN	33.14 ± 0.27	0.901 ± 0.005
NI	33.43 ± 0.22	0.916 ± 0.004

From Table 3, it can be seen that the SRCNN model performs the worst in terms of PSNR and SSIM metrics, with values of 30.24dB and 0.871, respectively.

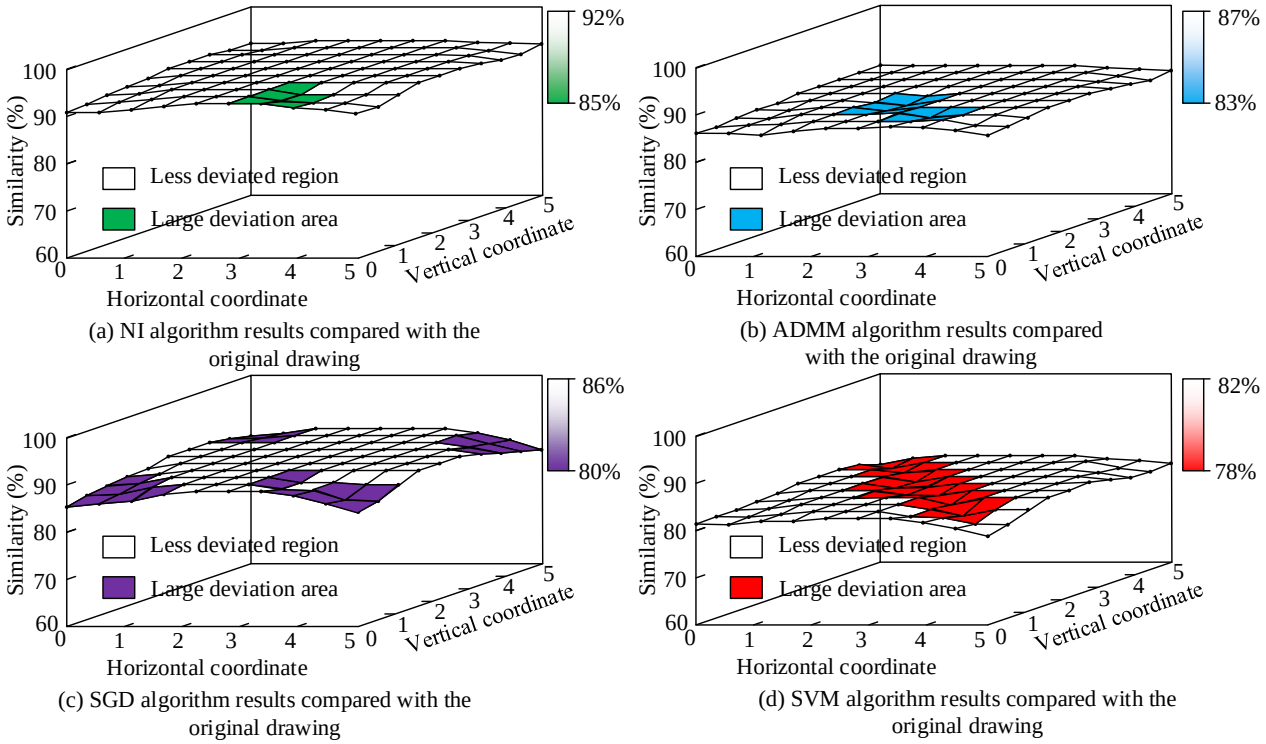
The NI algorithm achieves an average PSNR of 33.43 dB and SSIM of 0.916, higher than the other three models, with the smallest standard deviation. This indicates that the NI algorithm performs not only more accurately but also more stably, producing results closer to the original high-resolution image. In order to verify the working efficiency of the NI algorithm, 100 images from the DIV2K dataset were used to verify the processing speed of the algorithm, and the results are shown in Figure 7.



(a) Model processing time distribution (b) Each model average processing time

Figure 7: Comparison of single image processing time and average processing time

As shown in Figure 7(a) and Figure 7(b), the processing time of a single image using the NI algorithm did not exceed 20ms, with an average value of only 16ms. The processing time of a single image using the ADMM algorithm was between 20ms-30ms, with an average value of 25ms, the processing time of a single image using the SGD algorithm was between 30ms-35ms, with an average value of 32ms, the processing time of a single image using the SVM algorithm was between 40ms-50ms, with an average value of 44ms. This indicates that the NI algorithm is more efficient in image optimization. Subsequently, in order to verify the quality of graphics processed by the NI algorithm, each algorithm was used to optimize the images in the same test set, and the similarity between the processed images and the original images was compared. The results are shown in Figure 8.



(a) NI algorithm results compared with the original drawing

(b) ADMM algorithm results compared with the original drawing

(c) SGD algorithm results compared with the original drawing

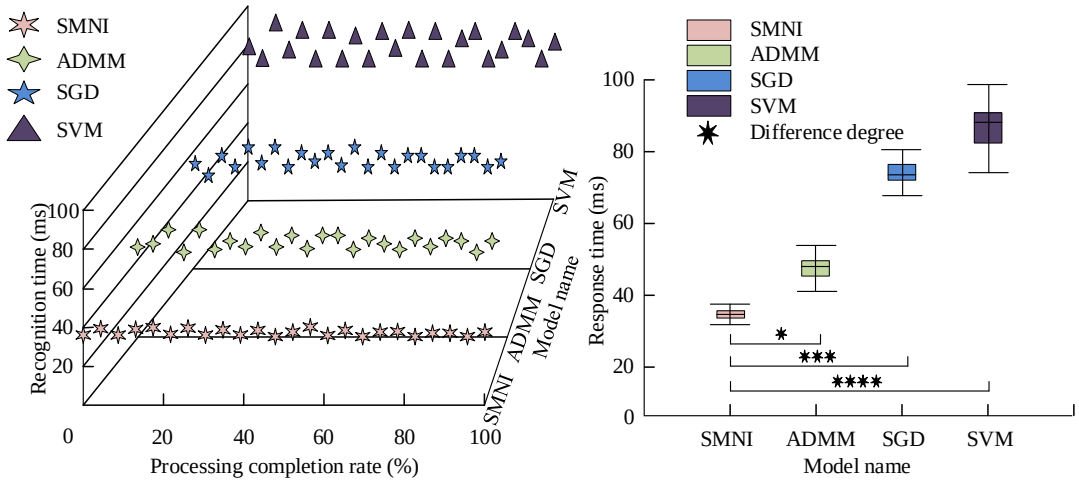
(d) SVM algorithm results compared with the original drawing

Figure 8: Comparison of processing results of the same image

As shown in Figure 8(a), the overall similarity of the image processed by the NI algorithm was 90.1%, and the similarity of each region of the image was relatively close, with only a small number of regions showing large similarity deviations. As shown in Figure 8(b), the overall similarity of the image processed by the ADMM algorithm was 86.3%, and the similarity of each region of the image was relatively close, with only a small number of regions showing large similarity deviations. As shown in Figure 8(c), the overall similarity of the image processed by the SGD algorithm was 85.4%, but the similarity distribution of each region of the image was not uniform, with the similarity in the four corners of the image being significantly lower than in the central region. As shown in Figure 8(d), the overall similarity of the image processed by the SVM algorithm was 80.1%, but the similarity deviation of each region of the image was large, with the similarity in the diagonal regions being significantly lower than that of other regions. It could be seen that the overall effect of the NI algorithm in image processing was better, and the processed image was more similar to the original image.

4.2 Verification of the actual application effect of the model based on NI and SR algorithm

Based on the verified NI algorithm, the study further verified the performance of the SMNI model. Real images were used as samples during the verification, and the model performance was evaluated based on the quality of the output images and the output time. The models composed of the ADMM algorithm, SGD algorithm, and SVM algorithm were selected for comparison. First, in order to verify whether the SMNI model had an advantage in processing speed, the same set of images was used to verify the processing efficiency. The results are shown in Figure 9.



(a) Response time of the model in the process

(b) Box diagram of the average response time

Figure 9: Comparison of processing time distribution of each model

As shown in Figure 9(a) and Figure 9(b), the image processing time distribution of the SMNI model was relatively concentrated, with a minimum of 36ms and a maximum of 40ms. The image processing time distribution of the ADMM model was relatively discrete, with the shortest processing time being 42ms and the longest being 57ms. The image processing time distribution of the SGD model was also relatively discrete, with a maximum of 80ms and a minimum of 64ms. The image processing time distribution of the SVM model was extremely wide, with the shortest processing time being 74ms and the longest being 97ms. It could be seen that the SMNI model was more efficient in image processing. Subsequently, in order to verify the image restoration ability of each model, each model was used to process the same real image divided into 100 small areas, and the structural similarity between the output image and the original image was compared. The results are shown in Figure 10.

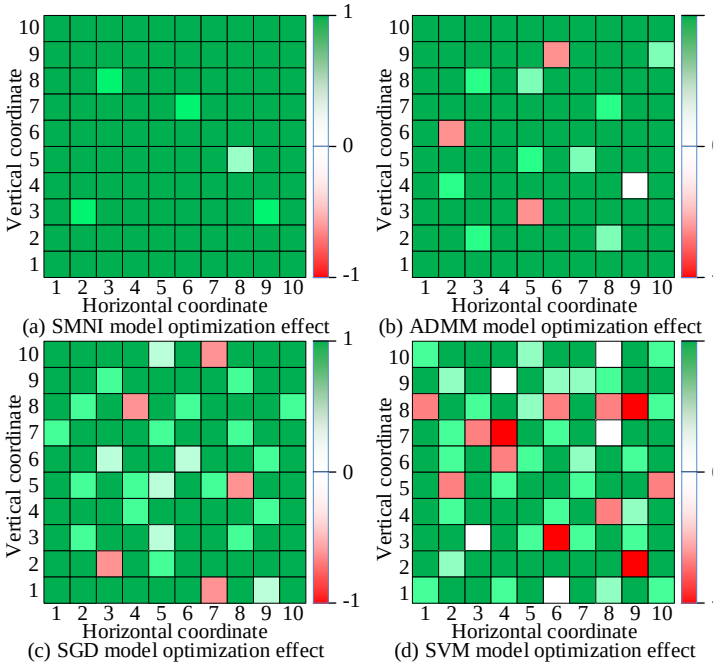


Figure 10: Comparison of model structure similarity preservation

As shown in Figure 10 (a), the structural similarity index of up to 95% of the local regions in the images processed by the SMNI model is extremely high (up to 0.997), indicating that the model has high restoration ability for these specific regions when performing image restoration. As shown in Figure 10 (b), 87% of the regions restored by the ADMM model exhibit high structural similarity, while 3% of the regions have relatively low similarity, indicating a high level of image restoration. As shown in Figure 10 (c), only 72% of the regions in the image processed by the SGD model exhibit high structural similarity, and 5% of the regions have structural similarity less than 0, indicating that the image restoration performance is moderate. As shown in Figure 10 (d), only 54% of the regions in the images processed by the SVM model exhibit high structural similarity, indicating poor image restoration. It could be seen that the SMNI model proposed in the study had higher accuracy when processing images. In order to verify the robustness of the model in the image generation process, Gaussian noise of different intensities was studied, and the proposed model was compared with classical adaptive median filtering (AMF) and Deep Neural Network for Image Denoising (DnCNN). The PSNR comparison results of three methods under different Gaussian noises are shown in Table 4.

Table 4: Comparison of PSNR of three methods under different gaussian noise

Noise variance	AMF	DnCNN	SMNI
0.01	26.2 dB	29.1 dB	29.3 dB
0.02	23.5 dB	26.6 dB	26.8 dB
0.03	21.4 dB	24.7 dB	24.7 dB
0.04	19.6 dB	23.0 dB	23.1 dB

From Table 4, it can be seen that at low variance (0.01), the PSNR index of the SMNI model is 29.3 dB, which is close to DnCNN and significantly better than AMF. The results indicate that the model has a certain ability to suppress additive Gaussian noise. To demonstrate the image processing effect of the proposed model more intuitively, the SMNI model, ADMM model, and SGD model were used to process the same blurred image. The image processing results are shown in Figure 11.

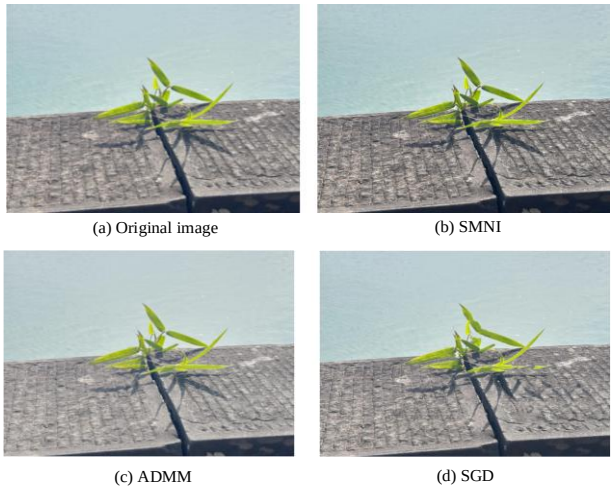


Figure 11: Image processing results

Comparing Figures 11 (a), (b), (c), and (d), the SMNI model produces images with high clarity, naturalness, and rich details, avoiding the excessive smoothing observed in ADMM and the distortion and artifacts seen in SGD.

4 Discussion

To improve the efficiency and accuracy of image processing models, this study proposes a NI algorithm that combines NAG with IBCD, and further integrates it with SR reconstruction and Mean Shift clustering to construct an SMNI image processing model. The results showed that the proposed NI algorithm achieved an average PSNR of 33.43 dB and SSIM of 0.916, higher than the SRCNN, EDSR, and RCAN models, with the smallest standard deviation. When processing images, the average processing time for a single image using the NI algorithm is 16ms, and the overall similarity of the images is 90.1%. The image processing time distribution of the SMNI model is relatively concentrated, with a minimum of 36ms and a maximum of 40ms. It achieves 100% structural similarity for 95% of the regions in the processed image compared to the original, with a maximum processing time per image of 40 ms. In summary, the SMNI model proposed in this study combines the NAG accelerated IBCD algorithm with a super-resolution reconstruction framework, demonstrating significant advantages in image processing efficiency and quality compared to related research. Compared with the traditional IBCD method proposed by Birgin and Mart í nez [6], the introduction of the NAG algorithm effectively accelerates the convergence process of block coordinate updates and reduces the number of iterations through its forward gradient mechanism. Compared with the generalized decentralized block optimization method proposed by Wang et al. [9], Mean Shift clustering significantly reduces the number of variable blocks that need to be independently optimized by intelligently grouping image blocks, thereby reducing computational complexity. This optimization strategy significantly improves the processing speed of the SMNI model while maintaining high image quality.

In practical applications, the proposed SMNI model is suitable for scenarios that require high image quality but relatively relaxed real-time requirements, such as medical image enhancement, remote sensing image analysis, and photography post-processing. The SMNI model has certain universality for different image types and resolutions, but parameter tuning is required for certain specific fields, such as medical imaging. The core value of the SMNI model lies in its ability to efficiently handle high-dimensional, non-convex optimization problems, making it potentially applicable to a wider range of complex dynamical system control tasks beyond image processing. For example, in the adaptive fuzzy control and output feedback projection lag synchronization of fractional order chaotic systems, there are significant nonlinearity and uncertainty in the system, and the real-time parameter tuning of the controller can be regarded as a dynamic optimization problem [21-22]. The proposed NAG accelerated IBCD method provides a new approach to solving such online optimization problems, and its block coordination mechanism can be used to adjust multiple controller parameters in parallel to meet the demanding timing requirements of real-time control. Although this work focuses on image processing, its advantages in computational efficiency and uncertainty handling provide a theoretical foundation for future applications in fields such as optimal control of nonlinear industrial processes or flexible robot trajectory planning [23-24]. However, in terms of computing resources, although SMNI models can be accelerated by GPUs, their computational intensity may be limited to deployment on resource constrained devices. Therefore, in practical applications, model performance and processing efficiency need to be balanced based on hardware conditions. Future work will focus on further optimizing clustering strategies and combining them with lightweight neural networks to further enhance the real-time performance and adaptability of image processing models. To evaluate the performance retention ability of the proposed SMNI model at different image resolutions, further testing was conducted on various specifications of images, including low resolution (such as 256 × 256), medium resolution (512 × 512), and high-resolution (1024 × 1024) images. The proposed SMNI model was integrated with current mainstream efficient image processing frameworks, including Feature-Affinity Based Knowledge Distillation (FAKD) and Linearly-Assembled Pixel-Adaptive Regression (LAPAR), and compared with the Information Multi-Distillation Network (IMDN). The performance comparison results of each model at different resolutions are shown in Table 5.

Table 5: Performance comparison of various models at different resolutions

Models	PSNR /dB	SSIM	Time (ms) 256×256	Time (ms) 512×512	Time (ms) 1024×1024
FAK	32.97	0.903	8.24	22.53	78.32
LAPAR	33.19	0.911	7.06	18.98	64.88
IMDN	33.26	0.913	9.35	25.07	88.04
SMNI	33.43	0.916	6.41	16.24	57.96

From Table 5, it can be seen that the SMNI model can maintain low processing time under different input sizes, and the PSNR and SSIM values are higher than the comparison model, which are 33.43 dB and 0.916, respectively. The results show that the SMNI model outperforms current advanced fast image optimization techniques in multiple efficiency and quality indicators, and its performance can be maintained at different resolutions, especially suitable for applications with high real-time requirements.

5 Conclusion

This study addressed the challenge that existing image processing techniques face in balancing processing efficiency and image quality. An innovative NI algorithm was proposed, which uses the NAG algorithm to improve

the convergence success rate of the IBCD algorithm. On this basis, an SMNI model was proposed, which optimizes image resolution parameters using the SR algorithm to achieve high-efficiency image processing. In summary, the SMNI model successfully integrates the advantages of the IBCD and NAG algorithms, meeting the requirements for processing efficiency, accuracy, and robustness. However, the study did not account for the effects of environmental factors such as lighting and haze when converting real images to digital signals. Future research will further validate the model's performance through more simulated experiments and continuously improve the model.

References

[1] Pal S, Roy A, Shivakumara P, Pal U. Adapting a Swin Transformer for License Plate Number and Text Detection in Drone Images. Artificial Intelligence and Applications, 2023, 1(3), 145-154, doi: 10.47852/bonviewAIA3202549.

[2] Jiang Z M, Rihan M, Deng Q, Mohamed E M, Khallaf H S, Wang X, Omar R. On the physical layer security of automotive dual-function radar-communication systems. IEEE Internet of Things Journal, 2023, 11(3): 5090-5104, doi: 10.1109/JIOT.2023.3301771.

[3] Attouch H, Fadili J. From the Ravine method to the Nesterov method and vice versa: a dynamical system perspective. SIAM Journal on Optimization, 2022, 32(3): 2074-2101, doi: 10.1137/22M1474357.

[4] Zhang B, Xiong W, Ma M, Wang M, Wang D, Huang X, Huang X. Super-resolution reconstruction of a 3 arc-second global DEM dataset. Science bulletin, 2022, 67(24): 2526-2530, doi: 10.1016/j.scib.2022.11.021.

[5] Rani R S, Madhavan P, Prakash A. Improving the Mean shift clustering algorithm for universal background model (UBM). Circuits, Systems, and Signal Processing, 2022, 41(7): 3882-3902, doi: 10.1007/s00034-022-01962-3.

[6] Birgin E G, Martínez J M. Block coordinate descent for smooth nonconvex constrained minimization. Computational Optimization and Applications, 2022, 83(1): 1-27, doi: 10.1007/s10589-022-00389-5.

[7] Rebegoldi S. Analysis of a variable metric block coordinate method under proximal errors. Annali Dell'universita'di Ferrara, 2024, 70(1): 23-61.

[8] Farias Maia L, Gutman D H. The randomized block coordinate descent method in the Hölder smooth setting. Optimization Letters, 2025, 19(1): 1-20, doi: 10.1007/s11565-022-00456-z.

[9] Wang Z, Bai M. An inexactly accelerated algorithm for nonnegative tensor CP decomposition with the column unit constraints. Computational Optimization and Applications, 2024, 88(3): 923-962, doi: 10.1007/s10589-024-00574-8.

[10] Van Cuong N, Hong Y W P, Sheu J P. UAV trajectory optimization for joint relay communication and image surveillance. IEEE Transactions on Wireless Communications, 2022, 21(12): 10177-10192, doi: 10.1109/TWC.2022.3182813.

[11] Wang S, Wang Z, Wang S, Ye Y. Deep image compression toward machine vision: A unified optimization framework. IEEE Transactions on Circuits and Systems for Video Technology, 2022, 33(6): 2979-2989, doi: 10.1109/TCSVT.2022.3230843.

[12] Hassan E, Shams M Y, Hikal N A, Elmougy S. The effect of choosing optimizer algorithms to improve computer vision tasks: a comparative study. Multimedia Tools and Applications, 2023, 82(11): 16591-16633, doi: 10.1007/s11042-022-13820-0.

[13] Sahu G, Seal A, Bhattacharjee D, Frischer R, Krejcar O. A novel parameter adaptive dual channel MSPCNN based single image dehazing for intelligent transportation systems. IEEE Transactions on Intelligent Transportation Systems, 2022, 24(3): 3027-3047, doi: 10.1109/TITS.2022.3225797.

[14] Frants V, Agaian S, Panetta K. QCNN-H: Single-image dehazing using quaternion neural networks. IEEE Transactions on Cybernetics, 2023, 53(9): 5448-5458, doi: 10.1109/TCYB.2023.3238640.

[15] Wang Y, Wan R, Yang W, Li H, Chau L P, Kot A. Low-light image enhancement with normalizing flow//Proceedings of the AAAI conference on artificial intelligence. 2022, 36(3): 2604-2612, doi: 10.1609/aaai.v36i3.20162.

[16] Chamain L D, Qi S, Ding Z. End-to-end image classification and compression with variational autoencoders. IEEE Internet of Things Journal, 2022, 9(21): 21916-21931, doi: 10.1109/JIOT.2022.3182313.

[17] Ren J, Yuan Y, Huang T, Jiang W, Feng W. Optimal design for the artificial-noise-aided IRS-MIMO-OFDM secure communications. IET Communications, 2022, 16(13): 1570-1581, doi: 10.1049/cmu2.12409.

[18] Sonntag K, Peitz S. Fast multiobjective gradient methods with Nesterov acceleration via inertial gradient-like systems. Journal of Optimization Theory and Applications, 2024, 201(2): 539-582, doi: 10.1007/s10957-024-02389-3.

[19] Du W, Tian S. Transformer and GAN-based super-resolution reconstruction network for medical images. Tsinghua Science and Technology, 2023, 29(1): 197-206, doi: 10.26599/TST.2022.9010071.

[20] Ameijeiras-Alonso J, Einbeck J. A fresh look at mean-shift based modal clustering. Advances in Data Analysis and Classification, 2024, 18(4): 1067-1095, doi: 10.1007/s11634-023-00575-1.

[21] Boulkroune A, Zouari F, Boubellouta A. Adaptive fuzzy control for practical fixed-time synchronization of fractional-order chaotic systems. Journal of Vibration and Control, 2025: 10775463251320258, doi: 10.1177/10775463251320258.

[22] Boulkroune A, Hamel S, Zouari F, Boukabou A,

Ibeas A. Output-Feedback Controller Based Projective Lag-Synchronization of Uncertain Chaotic Systems in the Presence of Input Nonlinearities. Mathematical Problems in Engineering, 2017, 2017(1): 8045803, doi: 10.1155/2017/8045803.

[23] Rigatos G, Abbaszadeh M, Sari B, Siano P, Cuccurullo G, Zouari F. Nonlinear optimal control for a gas compressor driven by an induction motor. Results in Control and Optimization, 2023, 11: 100226, doi: 10.1016/j.rico.2023.100226.

[24] Zouari F, Saad K B, Benrejeb M. Adaptive backstepping control for a single-link flexible robot manipulator driven DC motor//2013 International Conference on Control, Decision and Information Technologies (CoDIT). IEEE, 2013: 864-871, doi: 10.1109/CoDIT.2013.6689656.