Rendered Image Superresolution Reconstruction with Multichannel Feature Network

Zhipeng Ren 1, Jianping Zhao 1, Chunyi Chen 1, Yan Lou 1, Xiaocong Ma 2, and Pengyu Tao 1

1 School of Computer Science and Technology, Changchun University of Science and Technology, Changchun 130022, China
2 Muti-Media Communication Research Lab, University of Ottawa, Ottawa K1N 6N5, Canada

Correspondence should be addressed to Jianping Zhao; rzp973@gmail.com

Received 12 March 2022; Revised 23 June 2022; Accepted 20 July 2022; Published 29 August 2022

Academic Editor: Dongpo Xu

Copyright © 2022 Zhipeng Ren 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.

In the process of film and television production, clear images can give the audience a real sensory experience, but high-resolution images require a massive amount of production time and highly specialized imaging equipment, which is not a cost-effective solution at the moment. To achieve a better cost efficiency during video production, we propose a multichannel featured superresolution network model that utilizes rendered low-resolution images according to their characteristics. This model includes a feature extraction layer, a series of subnetworks, and a reconstruction module. Inside the network model, a series of subnetworks are cascaded to improve the information flow from coarse to fine, which helps to fully extract the depth, normal vector, edge, and texture features from low-resolution rendered images to reconstruct the high-resolution image. Additionally, residual learning is introduced at each stage to further improve the reconstruction performance. We experiment with the model on the classic Disney Monte Carlo datasets and compare it with several related algorithms. The results show that our algorithm is able to reconstruct the image with clearer details and texture. Thus, our research not only helps to maintain the audience’s sensory experience but also increases the efficiency of film and television production, which also brings considerable economic benefits.

1. Introduction

Image resolution is a set of performance parameters used to evaluate the richness of detailed information contained in an image. This includes temporal resolution, spatial resolution, and color scale resolution, which all together indicate the ability of an imaging system that can reflect the detailed information of an object. Compared with low-resolution (LR) images, high-resolution (HR) images usually contain higher pixel density, richer texture details, and higher fidelity. Modern film and television productions have high demands for the visual effect on picture qualities as technologies advance. However, to achieve a high-definition picture quality, it inevitably requires a long image rendering time, and the production cost also increases accordingly. How to reduce rendering time and production cost without affecting the visual experience of the audience is a vital research direction. More practically, it is important to maintain an ideal balance between the rendering time and the visual quality of an image such that the cost of the production can be kept minimum, and thus, how to improve the image’s resolution efficiently and quickly is the key to solving the above problems.

Physically, solving low-resolution problems of images is often too costly. Constrained by many factors such as processing power of the hardware, format type of the image, network bandwidth, and the image degradation model itself, it is hard to obtain an ideal high-resolution image without applying edge sharpening and block deblurring. In many cases, animations and effect maps are drawn by cartographic software to restore details. Similarly, it is entirely possible that the details of high-resolution (HR) images can be recovered from low-resolution (LR) images by using various types of algorithms. Early methods include interpolation, such as double edge interpolation and Lanczos resampling [1]. More powerful methods using statistical image prior
[2, 3] or internal patch recursion [4], probabilistic graphic model [5], neighborhood embedding [6, 7], sparse encoding [8], and linear or nonlinear regression are also used to convert LR images back into HR images [9–11].

The method of using deep learning to improve image resolution has made rapid development in recent years. Various superresolution network algorithms continue to be invented for better efficiency of restoring. In 2014, Dong et al. [12] first applied a convolutional neural network to an image superresolution task and proposed a superresolution convolutional neural network model to predict the nonlinear relationship between interpolated images and high-resolution images (SRCNN). The model carried out end-to-end image pair training, which greatly improved the reconstruction effect at that time and thus proved the excellent performance of deep learning in the field of image superresolution. On top of that, Dong et al. [13] proposed to enlarge the size of the deconvolution layer and improve the SRCNN model to reconstruct a high-resolution image (FSRCNN). In 2016, Kim et al. proposed two very deep convolution neural networks, VDSR [14] and DRCN [15] models, to deeply extract image features and to accelerate the network’s convergence speed by introducing the residual idea. It has been proved that the deep network can extract more features and achieve a better reconstruction effect. Tai et al. [16] constructed a persistent memory network for image restoration (MemNet), accompanied by memory blocks with a tight connection structure. In 2021, Hu et al. [17] designed a network to extract more detailed image features and cross-fused HR images (MSICF).

Although these models have made significant progress, they are mainly aimed at the restoration of natural images. However, there is a lack of research focused on the super-resolution reconstruction of rendered images with multi-channel features in the field of video production, and this has brought up drawbacks due to some bottlenecks. For example, the single-channel network cannot extract the main features such as depth and the normal vector of rendered images. Furthermore, the utilization rate of image information obtained from different channels is not sufficient, and this can result in insufficient details of reconstructed images. Given the above situation, we propose a super-resolution reconstruction method for rendered images and produce more realistic results. The main contributions of this paper are summarized as follows:

1. We propose a multichannel featured superresolution reconstruction algorithm of the rendered image, which is able to obtain multidimensional information from different channels such as depth, normal vector, edges, and texture in the rendered image. After the multichannel featured information passes through the network model, it is fused into three RGB color channels. At the same time, global residual and local residual learning are introduced to prevent the disappearance or explosion of the gradient, accelerate the convergence speed, and obtain a considerable reconstruction effect.

2. The network model includes a feature extraction layer, a series of subnetworks, and a reconstruction module. The extraction layer adopts double filters, which can retain high-frequency information such as image edge and texture details. The subnetworks cascade to improve the information flow from coarse to fine, and the reconstruction layer adopts weighting to improve the reconstruction accuracy. Compared with several other advanced algorithms, experiments show that the algorithm in this paper is more effective in image detail restoration.

3. In the process of video production, our innovative method firstly renders the low-resolution image and then applies the proposed superresolution network algorithm to reconstruct the high-resolution image. In this way, we not only maintain an excellent level of audience’s sensory experience but also reduce the film and television production time and cost for achieving increased economic benefits. As a result, this research has brought distinct economic and practical values to the video industry.

2. The Proposed Method

We propose a multichannel featured superresolution reconstruction algorithm for rendered images, as shown in Figure 1. It includes a feature extraction layer, a series of subnetworks, and a reconstruction module. The image input enters the subnetwork after convolution in the first layer, and the information is output to the reconstruction layer after passing through each subnetwork. At the same time, the overall residual learning is introduced. Finally, the reconstruction layer carries the weighted fusion to reconstruct the three-channel high-resolution image.

2.1. The Proposed Network. The extraction layer extracts the features of the original low-resolution image. The extraction layer of our design is composed of 64 filters with a convolution kernel size of 3 × 3. We defined convolution layer expression as \( S_0 = f_0(x) \), where \( x \) is the input low-resolution image, \( f_0 \) is the feature extraction function, and \( S_0 \) is the output after feature extraction. In the subnetwork shown in Figure 2, according to the multichannel features of the rendered image, we design a cross-module with multi-features to extract the multichannel, fuse the feature information, and then stack several submodules to learn the residual information between the input and output features. More specifically, the feature information of 10 channels enters the network, and the output of the feature extracted by convolution is used as the input of the next subnetwork. The convolution operation is carried out in a turn-taking sequence. Once the information has passed through all subnetworks, all types of features such as texture, depth, and normal vector can be fully extracted. At the same time, local residual learning is introduced to provide supplemental image feature data, which improves the accuracy and efficiency of image feature learning process. The first convolution layer adopts 64 filters, which are superimposed and
fused after entering the subnetwork; it is then followed by layers that adopt 128, 192, and 256 filters; all come with a size of $3 \times 3$, respectively.

In Figure 2, it can be represented by equation (1). We define $f_q(\cdot)$ as the feature extraction function and $S_{q-1}$ as the input of the $q^{th}$ subnetwork. $M$ is the number of subnetworks. The expression of subnetwork is $\{G_q^M(\cdot)\}_{M-1}^{1}$. We introduce local residual learning between $G(\cdot)$ and $G_{last}(\cdot)$ for the subnetwork to obtain different image features. We define $S_0$ as the expression of the first subnetwork, $ξ_q(\cdot)$ as the expression of the $q^{th}$ subnetwork, and $η$ as the number of subnetwork modules. Equation (1) can be expressed as follows:

$$S_q = G_q^M (G_{last}(\cdots G_q^3 (\cdots (G_q^1 (S_{q-1}) \cdots) \cdots) \cdots) + S_{q-1}.$$  

In the training process, to make the prediction closer to the real data, all prediction data are managed in a cascade way from the subnetwork to the final output of the whole network. We define $R_q(\cdot)$ as the expression of the reconstruction layer, $S_q$ as the output of the $q^{th}$ subnetwork, and $\tilde{y}$ as the output of the whole network. Concurrently, we use global residual learning to improve the reconstruction accuracy. In this way, all intermediate predictions are deeply convoluted, and the final output is weighted at the reconstruction layer. Equation (2) can be expressed as follows:

$$\tilde{y} = R_q(S_q) + x.$$  

2.2. Preliminary Work. For the preliminary preparation of data training, we select 200 HR images from the classic Disney Monte Carlo rendered image datasets which contain ten-channel information respectively, including "R," "G," "B," "normal.R," "normal.G," "normal.B," "albedo.R," "albedo.G," "albedo.B," and "depth.Z." The letter "R," "G," and "B" represent red, green, and blue, "normal.R," "normal.G," and "normal.B" represent three vectors perpendicular to the surface, and "albedo.R," "albedo.G," and "albedo.B" represent different albedo textures. The "depth.Z" represents the depth of the image. We expand the training data by performing horizontal, vertical, and flip operations on these images, reduce the sampling by 2, 3, and 4 times, and then restore them to their original size by applying a bicubic interpolation algorithm. Finally, the interpolated LR images and the original HR images are divided into $64 \times 64$ fragments, which are stored in pairs in a training data file.
2.3. Residual Learning. In previous neural network models, each layer learns a mapping of $y = F(x)$, where $x$ is the input, $y$ is the output, and $F(x)$ is the mapping function. One can think in the way that the error of the output becomes while, we add the original input at the end of each layer, of backpropagation as the error becomes larger. Meanwhile, we add the original input at the end of each layer, the output becomes $F(x) + x$, and the mapping relationship of each layer then becomes $y = F(x) + x$. It is obvious that the value of $F(x)$ tends to be irrelevant and negligible compared to the size of $x$ in this way, even if the number of layers continues to increase, the error $F(x)$ is still controlled at a very small value, as shown in Figure 3. Generally speaking, it is noteworthy that the residual learning includes global residual learning and local residual learning, which has been widely used in deep learning [18–20]. More particularly in this paper, our designed network model introduces global residual learning into the overall network and local residual learning into the subnetwork to ensure stable training can be carried out. Moreover, we also prove the effectiveness of residual learning by conducting subsequent training.

The depth of the whole model can be calculated by (3). On the right side inside the brackets, multiplication of 3 means that there are three convolution layers during one extraction operation, and the addition of 1 means that each subnetwork contains one convolution layer. In addition, the added value of 1 at the end of the right side represents the last convolution layer in the reconstructed network.

$$\text{Depth} = (M \times 3 + 1) \times n + 1. \quad (4)$$

3. Model Training

Every 64 groups of the stored training image data are divided into a batch, and each batch contains 64 * 64 segmented images that are categorized by 10 different channels. Once these images are prepared, they are then sent to the network as fuzzy input for training. After learning the deep convolution network and extracting the data features of each channel, we generate a 3-channel target image. By calculating the loss function between the target image and the real image and then updating the weight parameters through backpropagation, we can get an optimized network model. Through training, we discover that the deepening of the convolution layer of the subnetwork leads to a disappearance of the gradient, so we choose to set 10 convolution layers for each subnetwork. Tentatively, when the number of subnetworks is 3, the loss function can be reduced to an ideal effect, and this concludes that the depth of the whole network is also ideal when set to 31.

We compared the training-with-residual-learning results against those ones without it and found that the network with residual learning can accelerate the convergence and thus improve the accuracy. As shown in Table 1, we go through iteration 100 times, and the loss function converges to a very observable degree at the 40th epoch.

![Figure 3: Schematic diagram of the residual network.](image)

Table 1: Residual networks rapidly approach their convergence within 10 epochs.

| Scale | Epoch | 10 | 20 | 40 | 80 | 100 |
|-------|-------|----|----|----|----|-----|
| (a) Initial learning rate 0.1 | Residual | 36.06 | 36.85 | 37.25 | 37.98 | 37.95 |
| 2x Nonresidual | 36.70 | 33.82 | 36.41 | 37.31 | 37.07 |
| Difference 9.36 | 3.03 | 0.84 | 0.67 |
| (b) Initial learning rate 0.01. | Residual | 35.89 | 36.54 | 37.02 | 37.53 | 37.27 |
| 2x Nonresidual | 36.41 | 33.29 | 36.36 | 36.94 | 36.50 |
| Difference | 6.48 | 3.25 | 0.66 | 0.59 | 0.77 |
| (c) Initial learning rate 0.001. | Residual | 36.13 | 36.58 | 37.17 | 37.56 | 37.34 |
| 2x Nonresidual | 33.71 | 35.22 | 36.32 | 37.14 | 36.48 |
| Difference | 2.42 | 1.36 | 0.85 | 0.42 | 0.86 |

The bold values indicate the network with residual learning converges faster comparing without residual learning.

Our network is very deep and uses a 64 * 64 training map, which is larger than the receptive field information of general algorithms. Therefore, we used a stochastic gradient descent (SGD) solver and filled zero before convolution to keep the sizes of all feature maps the same for pixels near the image boundary [21]. We also train our network model at different scales, using 2x, 3x, and 4x images, and combine all LR and HR pairs of different scales into our training datasets. We set the momentum parameter to 0.9 and the weight attenuation to $10^{-4}$ [22, 23]. We also set the initial learning rate to 0.1, and then gradually reduced it by 10 times for every 25 epochs. To deal with the gradient problem of disappearance and explosion, we set the gradient clip on the gradient within a designed range $[-\theta, \theta]$, where $\theta$ is set to 0.4 [24–26]. Each convolution layer is followed by standard regularization and activation operations, and the negative slope of the activation layer is 0.2 [27, 28]. The training code of our network is shown in Table 2. After the training of 100 generations, we select the 83rd generation model as the follow-up experiment for analysis and evaluation according to the effect of the loss function.

4. Experiments and Comparisons

4.1. Environment and Data. We use a computer with Intel Core i7-3770 CPU (3.40 GHz), NVIDIA GeForce GTX
Table 2: The training code of our network.

| Step | Description |
|------|-------------|
| 1    | DataLoader (HR/LR) |
| 2    | Model = MFSR ()  |
| 3    | selected SGD () select MSEloss () |
| 4    | Loop Start |
| 4.1  | For \( i = 1, 2, \ldots, 100 \) |
| 4.2  | \( \text{calculate} \quad S_q \text{by equation (1)} \) |
| 4.3  | \( \text{calculate} \quad D_q \text{by equation (2)} \) |
| 4.4  | \( \text{calculate} \quad Y' \text{by equation (3)} \) |
| 4.5  | Lossfunction loss.backward () |
| 4.6  | update the parameter optimizer.step () |
| 4.7  | save network model |
| 4.8  | End for |
| 5    | Loop End |

1060 GPU, 16 GB memory, 64-bit Linux operating system, and 1 TB hard disk. The software includes Matlab R2016a, CUDA 8 tool kit, cuDNN 5.0, PyCharm, and PyTorch.

We then select 15, 25, 50, and 100 rendered images from the classic Disney Monte Carlo datasets as our test data. The test data are downsampled using Python program code by 2, 3, and 4 times and then magnified to the original size by applying bicubic interpolation method. Finally, the images are classified and stored in four folders named car, house, classroom, and bathroom, and the preparation of experimental data is completed.

4.2. Evaluation and Comparison. We take the peak signal-to-noise ratio (PSNR) [29], structural similarity (SSIM) [30], and execution time (ET) as the evaluation indexes of this experiment. The shorter the execution time (ET) is, the faster the algorithm runs. The mathematical expression of PSNR is shown in (5) and (6), where \( X_{\text{ori}} \) and \( X_{\text{res}} \) present the original image and the reconstructed image, respectively, the MSE refers to the mean squared error between \( X_{\text{ori}} \) and \( X_{\text{res}} \), and \( m \) and \( n \) represent the number of rows and columns of the image respectively. The unit of the PSNR is dB. It is known that the larger the value of the PSNR is, the closer the reconstructed image is to the ground truth image.

\[
\text{PSNR} = 10 \log_{10} \left( \frac{255^2}{\text{MSE}} \right), \tag{5}
\]

\[
\text{MSE} = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} (X_{\text{ori}(i,j)} - X_{\text{res}(i,j)})^2. \tag{6}
\]

The mathematical expression of the SSIM is shown in (7). The SSIM evaluates the image quality by comparing the similarity of the structural information between images. If the value of the SSIM is close to 1, it means that the reconstructed image is close to the original image.

\[
\text{SSIM} = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} \left( X_{\text{ori}(i,j)} \times X_{\text{res}(i,j)} \right)}{\sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} \left( X_{\text{ori}(i,j)} \right)^2} \times \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} \left( X_{\text{res}(i,j)} \right)^2}} \tag{7}
\]

We select this training model for experimenting with car, house, classroom, and bathroom test datasets. After inputting the experimental images into our network model and outputting the three-channel RGB super-resolution images, the color conversion is carried out, and the brightness values of the images are extracted as the reference values for calculating the evaluation indexes. The car datasets amplified by 2, 3, and 4 times produce an average result of the PSNR with values at 37.99 dB, 34.97 dB, and 32.83 dB and an average result of the SSIM with values of 0.9762, 0.9421, and 0.9179. To verify the effectiveness of the proposed method, the above experimental data are also input into other six advanced super-resolution algorithms, the SRCNN, the FSRCNN, the VDSR, the DRCN, the MemNet, and the MSICF, for direct results comparison.

As shown in Table 3, the PSNR value of the algorithm proposed in this paper exceeds the above reconstruction algorithms when using different combinations of magnifications and test sets. The experimental results show that our algorithm is better than the other six advanced super-resolution algorithms. On the house test set amplified twice, the PSNR value produced by the algorithm in this paper is 1.21 dB higher than the one produced by the MSICF and 1.87 dB higher than the one produced by the MemNet. On the bathroom test set, the PSNR value produced by the algorithm in this paper is 2.11 dB higher than the one produced by the VDSR and 3.28 dB higher than the one produced by the SRCNN. It is obvious in Table 4 that the SSIM values produced by the algorithm in this paper are higher than those produced by the other six advanced super-resolution algorithms when different magnifications and test sets are applied. With the classroom test set amplified by 4 times, the SSIM value produced by the algorithm in this paper is 0.0318 higher than the one produced by the MSICF and 0.0830 higher than the one produced by the MemNet. On the car test set, the SSIM value produced by the algorithm in this paper is 0.0321 higher than the one produced by the VDSR and 0.0505 higher than the one produced by the SRCNN. On the house test set amplified twice, the SSIM value produced by the algorithm in this paper is 2.11 dB higher than the one produced by the VDSR and 3.28 dB higher than the one produced by the SRCNN. On the bathroom test set amplified by 4 times, the SSIM value produced by the algorithm in this paper is 0.0318 higher than the one produced by the MSICF and 0.0830 higher than the one produced by the MemNet.
Table 3: Average PSNR for different algorithms at car, house, classroom, and bathroom.

| Dataset  | Scale | Bicubic | SRCNN [12] | FSRCNN [13] | VDSR [14] | DRCN [15] | MemNet [16] | MSICF [17] | Ours  |
|----------|-------|---------|------------|-------------|-----------|-----------|-------------|----------|-------|
| Car      | x2    | 32.02   | 35.01      | 35.49       | 36.29     | 36.35     | 36.55       | 36.77    | 37.99 |
| House    | x2    | 27.86   | 30.55      | 30.98       | 31.74     | 31.83     | 32.01       | 32.67    | 33.88 |
| Classroom| x2    | 24.42   | 26.28      | 26.47       | 27.15     | 27.16     | 27.08       | 28.75    | 29.72 |
| Bathroom | x2    | 27.39   | 30.06      | 30.35       | 31.23     | 31.32     | 31.46       | 32.58    | 33.34 |
| Car      | x3    | 28.60   | 31.59      | 32.07       | 32.91     | 33.16     | 33.34       | 33.92    | 34.97 |
| House    | x3    | 23.51   | 25.62      | 25.71       | 26.83     | 26.74     | 26.96       | 27.35    | 28.51 |
| Classroom| x3    | 22.31   | 24.09      | 24.18       | 25.13     | 25.10     | 25.06       | 26.37    | 27.65 |
| Bathroom | x3    | 23.05   | 25.11      | 25.20       | 26.54     | 26.43     | 26.37       | 26.84    | 28.01 |
| Car      | x4    | 26.80   | 29.81      | 30.23       | 31.17     | 31.25     | 31.46       | 31.88    | 32.83 |
| House    | x4    | 21.72   | 23.34      | 23.45       | 24.57     | 24.53     | 24.66       | 25.17    | 26.48 |
| Classroom| x4    | 20.05   | 21.77      | 21.92       | 22.84     | 22.81     | 22.84       | 23.72    | 24.91 |
| Bathroom | x4    | 21.48   | 23.03      | 23.12       | 24.26     | 24.18     | 24.35       | 24.88    | 26.03 |

The bold values indicate the best performance.

Table 4: Average SSIM for different algorithms at car, house, classroom, and bathroom.

| Dataset  | Scale | Bicubic | SRCNN [12] | FSRCNN [13] | VDSR [14] | DRCN [15] | MemNet [16] | MSICF [17] | Ours  |
|----------|-------|---------|------------|-------------|-----------|-----------|-------------|----------|-------|
| Car      | x2    | 0.8978  | 0.9348     | 0.9445      | 0.9519    | 0.9531    | 0.9603      | 0.9610   | 0.9762 |
| House    | x2    | 0.8503  | 0.9131     | 0.9127      | 0.9115    | 0.9210    | 0.9298      | 0.9349   | 0.9457 |
| Classroom| x2    | 0.7341  | 0.7998     | 0.8085      | 0.8292    | 0.8275    | 0.8283      | 0.8649   | 0.8916 |
| Bathroom | x2    | 0.8473  | 0.8971     | 0.9012      | 0.9101    | 0.9074    | 0.9113      | 0.9337   | 0.9406 |
| Car      | x3    | 0.8124  | 0.8869     | 0.8985      | 0.9190    | 0.9205    | 0.9243      | 0.9284   | 0.9421 |
| House    | x3    | 0.6583  | 0.7579     | 0.7611      | 0.8035    | 0.7998    | 0.8073      | 0.8229   | 0.8544 |
| Classroom| x3    | 0.6173  | 0.7225     | 0.7244      | 0.7513    | 0.7529    | 0.7531      | 0.7917   | 0.8382 |
| Bathroom | x3    | 0.6288  | 0.7432     | 0.7435      | 0.7966    | 0.7968    | 0.7851      | 0.8015   | 0.8293 |
| Car      | x4    | 0.7127  | 0.8674     | 0.8852      | 0.8858    | 0.8863    | 0.8871      | 0.8898   | 0.9179 |
| House    | x4    | 0.5903  | 0.6551     | 0.6572      | 0.6779    | 0.6784    | 0.6795      | 0.7411   | 0.7729 |
| Classroom| x4    | 0.5138  | 0.5914     | 0.6015      | 0.6671    | 0.6679    | 0.7134      | 0.7646   | 0.7964 |
| Bathroom | x4    | 0.5715  | 0.6478     | 0.6499      | 0.6685    | 0.6674    | 0.7236      | 0.7612   | 0.7612 |

The bold values indicate the best performance.

Figure 4: Continued.
4.3. Experimental Analysis. We analyze the above algorithms in terms of sensory effects. As shown in Figures 6–9, we compare our algorithm with the current six advanced algorithms. Our method can clearly reconstruct the edges, lines, and textures of the image and thus improve the quality of the image by doing so. The experimental results on the "60272340–08192spp" picture in the bathroom test set enlarged by 3 times are shown in Figure 6. The image restored by
our method shows a better visual experience with improved clarity of the background. The gloss and smoothness of the two cylinders have also been significantly improved, while other algorithms still have various degrees of blur or distortion.

The experimental results on the “16532963–08192spp” picture in the car test set enlarged by 3 times are shown in Figure 7. The image recovered by our method is closer to the real image. The outline of the car rearview mirror is relatively complete and smooth, and the lines, edges, and corners of the car frame are clearly visible. The recovery effect of the VDSR algorithm is obviously blurred, and the images recovered by the MemNet and the MSICF algorithms are too stiff in terms of contour and lines.

The effect of the experiment on the “98036695–08192spp” picture in the 4x enlarged classroom test set is shown in Figure 8. The SRCNN and the DRCN algorithms...
blur and distort the restoration of window frame lines, and the MemNet, the MSICF, and other algorithms distort the edge of the flag. Our method can restore clear lines and visible edge of the flag and make vertical lines and corners of the wall behind the flag to look more intuitively at the same time.

The effect of the experiment on the "32481698–08192pp" picture in the 4x enlarged house test set is shown in Figure 9. Most algorithms such as the FSRCNN and the DRCN do not restore enough clarity to the lines and gaps of the fence, resulting in distortion and deformation. Our method can give a clear view on the
number and gap of fences, and thus, our restoration on edges, corners, and cylinders is obviously closer to the real image.

5. Conclusion

In this paper, the main contributions are as follows: (1) Through the above experiments, it is proved that our method is able to extract more features of the image and reconstruct the image with clearer details and texture than other related algorithms. (2) Film and television producers can save time on rendering images and then utilize our proposed model to superresolution the images later. This provides an excellent cost-effective solution that may increase the profitability to the current video industry. (3) Our method does not affect the audience’s sensory experience when LR images come to play. This is a major advantage when data size is limited. The research content of this paper provides an initial solution to the stated problem with promising results. More specifically, we have achieved the expected research objectives, have reached the goal of creating a cost-effective solution to the film and television production without affecting the audience’s sensory experience, and thus have brought distinct economic values to the industry. In the future, we will continue to optimize the parameters of the network model and to introduce the attention mechanism into the model to improve the efficiency and recovery effect of the algorithm. At the same time, we will also apply our research results to the field of medical CT imaging and infrared imaging with remote sensing.

Data Availability

The dataset can be downloaded at the website: https://studios.disneystudios.com/download-deep-learning-denoising-data-set/, password: DRS_2008_Data1.

Ethical Approval

The study complies with ethical standards.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Zhipeng Ren contributed to conceptualization, methodology, and editing. Jianping Zhao contributed to supervision and review. Chunyi Chen contributed to validation. Yan Lou contributed to training and testing. Xiaocong Ma contributed to testing and review. Pengyu Tao contributed to writing, editing, and software.

Acknowledgments

This work was supported by the 2022 International Cooperation Project of Jilin Province Science and Technology Department (GH010290892), the Jilin Provincial Education Department (JJKH20220774KJ), and the State Key Laboratory of Applied Optics (SKLAO2022001A07).

References

[1] C. E. Duchon, "Lanczos filtering in one and two dimensions," Journal of Applied Meteorology, vol. 18, no. 8, pp. 1016–1022, 1979.
[2] K. I. Kim and Y. Kwon, "Image Single-Image Super-Resolution Using Sparse Regression and Natural Image Prior," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 6, pp. 1127–1133, 2010.
[3] J. Sun, Z. Xu, and H. Y. Shum, "Image super-resolution using gradient profile prior," in Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, June 2008.
[4] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in proceedings of the 2009 IEEE 12th International Conference on Computer Vision, pp. 1–8, Kyoto, Japan, September 2009.
[5] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael, "Learning low-level vision," International Journal of Computer Vision, vol. 40, no. 1, pp. 25–47, 2000.
[6] H. Chang, D. Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004, pp. 275–282, Washington, DC, USA, June 2004.
[7] X. Chen and C. Qi, "Nonlinear neighbor embedding for single image super-resolution via kernel mapping," Signal Processing, vol. 94, pp. 6–22, 2014.
[8] J. Jianchao Yang, J. Wright, T. S. Huang, and Y. Yi Ma, "Image super-resolution via sparse representation," IEEE Transactions on Image Processing, vol. 19, no. 11, pp. 2861–2873, 2010.
[9] H. Wang, X. Gao, K. Zhang, and J. Li, "Fast single image super-resolution using sparse Gaussian process regression," Signal Processing, vol. 134, pp. 52–62, 2017.
[10] R. Timofte, V. D. Smet, and L. V. Gool, "A+: adjusted anharmonic neighborhood regression for fast super-resolution," in Proceedings of the 2th Asian Conference on Computer Vision, pp. 111–126, Singapore, November 2014.
[11] S. Schulter, C. Leistner, and H. Bischof, "Fast and accurate image upsampling with super-resolution forests," in Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3791–3799, Boston, MA, USA, June 2015.
[12] C. Dong, C. C. Loy, K. He, X. Tang, Learning a Deep Convolutional Network for Image super-resolution," in Proceedings of the 13th European Conference on Computer Vision, pp. 184–199, Zurich, Switzerland, September 2014.
[13] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in Proceedings of the 14th European Conference 2016, pp. 391–407, Amsterdam, The Netherlands, October 2016.
[14] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1646–1654, Las Vegas, NV, USA, June 2016.
[15] J. Kim, K. K. Lee, and K. M. Lee, "Deeply-reconvolutional network for image super-resolution," in Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1637–1645, Las Vegas, NV, USA, June 2016.
[16] Y. Tai, J. Yang, X. Liu, and C. Xu, “MemNet: a persistent memory network for image restoration,” in Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), pp. 4549–4557, Venice, Italy, October 2017.

[17] Y. Hu, X. Gao, J. Li, Y. Huang, and H. Wang, “Single image super-resolution with multi-scale information cross-fusion network,” Signal Processing, vol. 179, Article ID 107831, 2021.

[18] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778, Las Vegas, NV, USA, June 2016.

[19] M. Haris, G. Shakhnarovich, and N. Ukita, “Deep back-projection networks for super-resolution,” in Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1664–1673, Salt Lake City, UT, USA, June 2018.

[20] J. Yamanaka, S. Kuwashima, and T. Kurita, “Fast and accurate image super resolution by deep CNN with skip connection and network in network,” in Proceeding of the 2017 24th International Conference on Neural Information Processing, pp. 217–225, Guangzhou, China, November 2017.

[21] C. Ledig, L. Theis, F. Huszar et al., “Photo-realistic single image super-resolution using a generative adversarial network,” in Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 105–114, Honolulu, HI, USA, July 2017.

[22] X. Yang, T. Xie, L. Liu, and D. Zhou, “Image super-resolution reconstruction based on improved Dirac residual network,” Multidimensional Systems and Signal Processing, pp. 1–18, 2021.

[23] A. Esmaeilzehi, M. O. Ahmad, and M. N. S. Swamy, “SRNSSI: a deep light-weight network for single image super resolution using spatial and spectral information,” IEEE Transactions on Computational Imaging, vol. 7, pp. 409–421, 2021.

[24] Y. Wang, F. Perazzi, R. McWilliams, A. Sorkine-Hornung, O. Sorkine-Hornung, and C. Schroers, “A fully progressive approach to single-image super-resolution,” in Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 977–986, Salt Lake City, UT, USA, June 2018.

[25] R. K. Srivastava, K. Greff, and J. Schmidhuber, “Training very deep networks,” in Proceedings of the 28th International Conference on Neural Information Processing Systems, pp. 2377–2385, Montreal Canada, December 2015.

[26] Y. Chen, L. Liu, V. Phonevilay et al., “Image super-resolution reconstruction based on feature map attention mechanism,” Applied Intelligence, vol. 51, no. 7, pp. 4367–4380, 2021.

[27] J. W. Soh, S. Cho, and N. I. Cho, “Meta-transfer learning for zero-shot super-resolution,” in Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3513–3522, Seattle, WA, USA, June 2020.

[28] R. Timofte, R. Rothe, and L. V. Gool, “Seven ways to improve example-based single image super resolution,” in Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1865–1873, Las Vegas, NV, USA, June 2016.

[29] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600–612, 2004.

[30] H. R. Sheikh, A. C. Bovik, and G. de-Veciana, “An information fidelity criterion for image quality assessment using natural scene statistics,” IEEE Transactions on Image Processing, vol. 14, no. 12, pp. 2117–2128, 2005.

[31] Y. Chen, V. Phonevilay, J. Tao, X. Chen, R. Xia, Q. Zhang, K. Yang, J. Xiong, J. Xie, The face image super-resolution algorithm based on combined representation learning,” Multimedia Tools and Applications, vol. 80, no. 20, pp. 30839–30861, 2021.