electronics

MDPI

Article

Deep Convolutional Dictionary Learning Denoising Method Based on Distributed Image Patches

Luqiao Yin [1], Wenqing Gao [1] and Jingjing Liu [1,2,*]

[1] Shanghai Key Laboratory of Chips and Systems for Intelligent Connected Vehicle, School of Microelectronics, Shanghai University, Shanghai 200444, China; lqyin@shu.edu.cn (L.Y.); 21723698@shu.edu.cn (W.G.)
[2] State Key Laboratory of Integrated Chips and Systems, Fudan University, Shanghai 201203, China
* Correspondence: jjliu@shu.edu.cn (J.L.)

Abstract: To address susceptibility to noise interference in Micro-LED displays, a deep convolutional dictionary learning denoising method based on distributed image patches is proposed in this paper. In the preprocessing stage, the entire image is partitioned into locally consistent image patches, and a dictionary is learned based on the non-local self-similar sparse representation of distributed image patches. Subsequently, a convolutional dictionary learning method is employed for global self-similarity matching. Local constraints and global constraints are combined for effective denoising, and the final denoising optimization algorithm is obtained based on the confidence-weighted fusion technique. The experimental results demonstrate that compared with traditional denoising methods, the proposed denoising method effectively restores fine-edge details and contour information in images. Moreover, it exhibits superior performance in terms of PSNR and SSIM. Particularly noteworthy is its performance on the grayscale dataset Set12. When evaluated with Gaussian noise $\sigma = 50$, it outperforms DCDicL by 3.87 dB in the PSNR and 0.0012 in SSIM.

Keywords: micro-LED display; distributed image patches; deep convolutional dictionary learning; confidence-weighted fusion technique

Citation: Yin, L.; Gao, W.; Liu, J. Deep Convolutional Dictionary Learning Denoising Method Based on Distributed Image Patches. *Electronics* **2024**, *13*, 1266. https://doi.org/10.3390/electronics13071266

Academic Editor: Stefanos Kollias

Received: 22 February 2024
Revised: 19 March 2024
Accepted: 27 March 2024
Published: 28 March 2024

1. Introduction

Micro-LED displays are pivotal in human–computer interaction and are renowned for their brightness and contrast [1,2]. They are indispensable in military applications like helmet displays and tactical goggles [3–5] and have revolutionized civilian sectors, enhancing automotive displays and medical imaging [6–8]. Despite their versatility, challenges during manufacturing, including thermal, optical, and Gaussian noise, compromise image quality [9–11]. Thus, effective denoising methods are crucial for optimal visual clarity. Traditional image denoising methods include filtering-based denoising methods, which smooth the image to reduce the impact of noise. There are also some traditional denoising methods based on the sparse representation theory of images, which remove noise by finding sparse representations of images under appropriate dictionaries [12,13]. With the continuous advancement of science and technology, denoising methods based on deep learning have been continuously proposed. Among them, deep learning models are used to denoise images through end-to-end learning. There are also generative adversarial networks that learn to generate noise-free images, resulting in more realistic images [14–16].

In addressing these challenges, traditional denoising methods offer foundational approaches and can be roughly divided into two categories [12,15]. The first category is based on the filtering principle, including various noise reduction methods for different image processing scenarios. Among them, bilateral filtering [17] stands out as a spatial method which can reduce noise and effectively maintain edges. Gaussian filtering [18] is a common method of traditional image denoising; using a Gaussian filter to smooth the image can suppress the high-frequency noise in the image. Additionally, wavelet transform [19] excels

at capturing both high- and low-frequency components, making it versatile for various applications. In addition, block-matching 3D (BM3D) filtering [20] is a collaborative filtering approach which uses the similarities in image patches to enhance the denoising effect. The second category is based on sparse representation theory and includes various denoising methods that improve image restoration by using mathematical frameworks. One prominent method within this category is K-singular value decomposition (K-SVD) [21]. K-SVD effectively captures the basic features of images by decomposing them into sparse representations. Another noteworthy method is total variational (TV) [22] denoising, which focuses on minimizing the total variation of the image, preserving its structural details and reducing noise. Weighted nuclear norm minimization (WNNM) [23], another method in this category, combines sparse representation with nuclear norm minimization, thus enhancing its ability to handle complex noise patterns. Although these traditional denoising methods exhibit good performance in certain image processing scenarios, they usually need fine parameter adjustment and rely on prior information constructed manually.

In addition to traditional denoising methods, recent advancements in deep learning have introduced novel approaches that leverage the power of neural networks for effective noise reduction in Micro-LED displays. Compared with traditional methods, deep learning-based approaches do not require pre-made assumptions and can automatically learn network parameters [14,24,25]. Therefore, they are widely used in image denoising. Zhang et al. [18] proposed a simple and effective denoising convolutional neural network (DnCNN), which enhances denoising performance by overlapping convolutional layers, batch normalization, and residual networks. In the same year, they introduced an iterative residual convolutional neural network (IRCNN) [26], employing multiple iterations for image denoising. Zhang et al. [27] introduced a fast and flexible denoising network (FFDNet). By learning the distribution and characteristics of image noise, the adaptability and robustness of neural networks to different levels of noise were enhanced. Alsaiari et al. [28] proposed a generative adversarial network (GAN) by using a small number of samples for rendering and passing the noisy image to the network, which generates high-quality realistic images. Im et al. [29] introduced a denoising approach utilizing variational autoencoder (VAE), which learns to generate a noise-free rendition of an image, thereby simulating noise and producing a clean image. Simon et al. [30] proposed the compressed sparse column (CSC) with rotation-invariant convolution, a method that employs convolutional neural networks to learn sparse coding constraints. Scetbon et al. [31] introduced an end-to-end deep method based on K-singular value decomposition called learned K-singular value decomposition (DK-SVD). They replaced the prior of L_0 in K-SVD with the prior of L_1 and unfolded the sparse coding through the iterative shrinkage thresholding algorithm (ISTA) [32]; the dictionary was parameterized by using a multi-layer perceptron (MLP) module. Subsequently, Zheng et al. [33] proposed the deep convolutional dictionary learning (DCDicL) denoising method, which simultaneously learns deep priors for both dictionaries and coefficients. While these methods have made progress in adaptability, robustness, the understanding of complex structures, and denoising effectiveness, most deep denoising methods still face challenges such as a large number of parameters, high complexity, and weak robustness.

Inspired by both traditional K-SVD and DCDicL methods, the current trend in research is to integrate the strengths of each approach while mitigating their respective limitations. The traditional K-SVD method excels in mathematical definitions and interpretability, whereas the DCDicL method shines in adapting to complex data structures and handling large-scale datasets. This paper proposes a deep convolutional dictionary learning denoising method based on distributed image patches for Micro-LED displays, as shown in Figure 1. At first, the whole image is divided into locally consistent distributed image patches. Then, non-local self-similar sparse representation dictionary learning is applied to distributed image patches, capturing their detailed structural characteristics. Subsequently, a deep convolutional network is used to match the global self-similarity. Finally, the denoising results of small image patches are merged through the weighted

superposition of confidence evaluations. The proposed method makes full use of the advantages of distributed image patches, dictionary learning, and convolutional sparse coding structures and overcomes the limitations of the artificial prior. The experimental results demonstrate that the method proposed in this paper achieves significant improvement in image denoising, highlighting its potential contribution to the field of Micro-LED display technology. The main contributions of this paper are listed below.

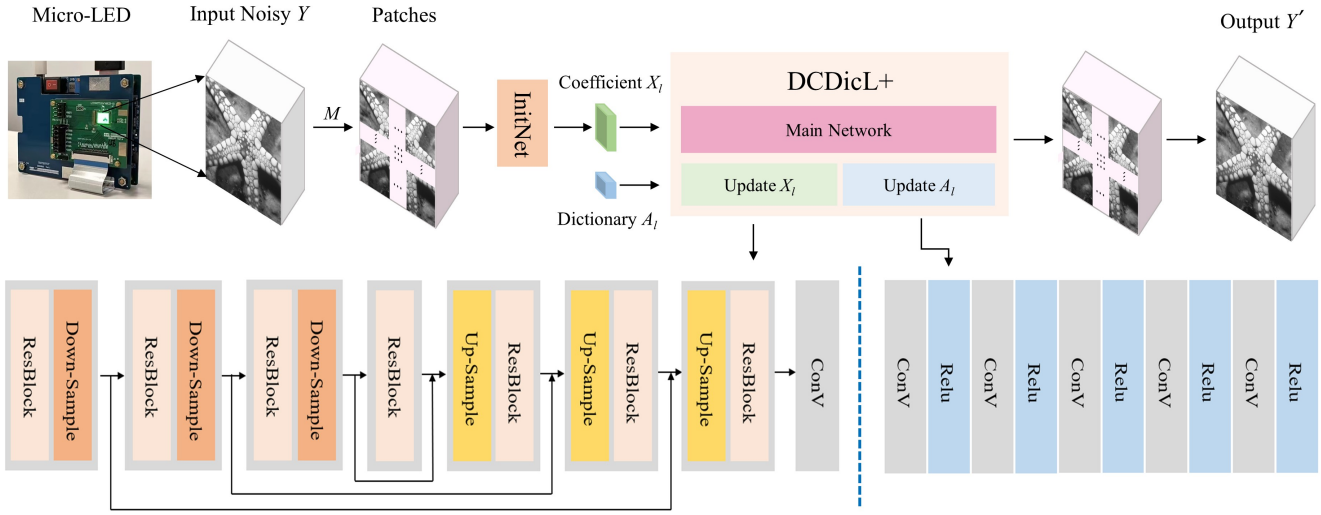


Figure 1. The overall structure of Dcidicl+.

- **Novel denoising method.** An improved denoising method based on deep convolutional dictionary learning, where images are decomposed into small patches and adaptive dictionary learning is conducted, is proposed. This method provides a better representation of the fine structure of images, as both global and local features are incorporated, allowing for a more comprehensive representation of the fine structure present in images.
- **Optimization algorithm.** A new confidence-weighted fusion algorithm is developed to optimize the proposed method in our model; it can utilize the structures of the convolutional sparse coding structure, and local and global operators.
- **Different applications.** On the grayscale dataset Set12, compared with DCDicL, the PSNR and SSIM of the proposed method are increased by 3.87 dB and 0.0012 at the noise level of $\sigma = 50$. On the color dataset Urban100, compared with DCDicL, the PSNR and SSIM of the proposed method are increased by 3.49 dB and 0.0133 at the noise level of $\sigma = 50$.

The structure of this paper is as follows: Section 2 provides an extensive review of the relevant literature. In Section 3, we introduce our denoising method in detail. Comparative experiments demonstrating the effectiveness of the proposed denoising method are presented in Section 4. Finally, Section 5 summarizes the research results of this paper.

2. Related Works

2.1. Patch Denoising

Patch denoising is an image denoising method which eliminates or reduces noise around the modeling and processing of small patches in the image. The motivation behind patch denoising is to use the correlation between adjacent pixels in an image, aiming at restoring the real information of the image more accurately through the modeling of local patches. It is worth noting that the deep K-SVD (DK-SVD) method, which is integrated with deep learning, is an outstanding example in this category.

The DK-SVD method redesigns a learnable architecture based on K-SVD. Suppose that $x \in \mathbb{R}^{\sqrt{m}\times\sqrt{m}}$ is the input clean image patch. We define x = Aα, where $A \in \mathbb{R}^{m\times n}$ denotes a dictionary atom and $\alpha \in \mathbb{R}^n$ denotes the sparse vector. The noisy patch $y \in \mathbb{R}^{\sqrt{m}\times\sqrt{m}}$ is degraded by additive white Gaussian noise (AWGN) with standard deviation σ. Unlike K-SVD, DK-SVD uses the l_1-norm instead of the l_0-norm, replacing the greedy algorithm with the l_1-based iterative soft-thresholding algorithm (ISTA) [32]. This objective can be formulated as

$$\hat{\alpha} = \arg\min_{\alpha} \frac{1}{2}\|A\alpha - y\|_2^2 + \lambda\|\alpha\|_1, \tag{1}$$

where λ is the regularization coefficient and $\|\cdot\|_1$ denotes the sum of the absolute values of the vector, which is the l_1-norm.

DK-SVD employs the iterative soft-thresholding algorithm (ISTA) to learn sparse matrix $\hat{\alpha}$, which can be represented by the following iterative formula:

$$\hat{\alpha}_{k+1} = S_{\lambda/c}(\hat{\alpha}_k - \frac{1}{c}A^T(A\hat{\alpha}_k - y)); \hat{\alpha}_0 = 0, \tag{2}$$

where k is the number of iterations, c denotes the square spectral norm of A, and $S_{\lambda/c}$ represents the soft-thresholding operator function.

$$[S_\mu(w)]_i = sgn(w_i)(|w_i| - \mu)_+, \tag{3}$$

As illustrated in Formula (3), by utilizing the proximal gradient descent approach, the sparse coding component is transformed into a trainable version. Performing operations on each patch of the image is essentially equivalent to convolving the image.

To control the error in a manageable way, a separate parameter, λ_k, is learned for each y_k. A multi-layer perceptron (MLP) network is utilized to learn a regression function, which maps inputs to outputs in the following manner: $\lambda = f_\theta(y)$, where θ represents the parameters of the MLP network.

During the patch reconstruction phase, the denoised image patch matrix ($\hat{y}$) is obtained. Based on the already determined dictionary A and sparse vector $\hat{\alpha}$, the denoised image patch can be reconstructed as follows: $\hat{y} = A\hat{\alpha}$, where A represents a set of learnable parameters. The entire patch denoising and patch reconstruction process is highly correlated with the convolutional sparse coding approach.

The complete end-to-end method architecture can be described as follows: The input image is segmented into fully overlapping patches at first; then, each corrupted patch is processed through the aforementioned patch denoising stage, and finally, the denoised versions of these patches are averaged to reconstruct the image. In the final step, the original K-SVD is abandoned, and a learnable method is designed to reconstruct the image. Let $u \in \mathbb{R}^{\sqrt{m}\times\sqrt{m}}$ be the weight coefficient for each patch; the reconstructed image is obtained by the following formula:

$$\hat{X} = \frac{\sum_k R_k^T(u \odot \hat{\alpha}_k)}{\sum_k R_k^T u}, \tag{4}$$

where $\odot$ is the Schur product. In the entire algorithm, the parameters that can be learned are θ (the vector of the parameters of the MLP), c (the step size in the ISTA), A (the dictionary), and u (the average weight of patches).

The main contribution of DK-SVD is to propose an end-to-end framework for deep learning methods. The framework retains the original computing path of K-SVD and redesigns a framework based on supervised learning. This architecture needs fewer learning parameters and retains the essence of K-SVD. Compared with the classical K-SVD method, its performance is significantly improved, and it is very close to the most advanced denoising method based on deep learning.

However, DK-SVD still uses fixed priors and adopts a generic dictionary rather than an image adaptive dictionary. In addition, the performance of the DK-SVD method is still behind that of many deep learning methods.

2.2. Convolutional Dictionary Learning

Convolutional dictionary learning denoising is a signal processing- and machine learning-based method employed for handling images contaminated by noise [34,35]. This method includes two key steps: dictionary learning and denoising processing.

In the initial stage, dictionary learning represents the first step of convolutional dictionary learning denoising. During this stage, the method endeavors to learn a set of fundamental structures or features from the input image, forming a dictionary. This dictionary encompasses convolutional kernels capable of efficiently representing the content of the image. The objective of dictionary learning is to compactly represent the input image, ensuring that the dictionary can effectively capture the local structures within the image.

Following the acquisition of the dictionary, the input image can undergo sparse representation using the dictionary. Sparse representation means that the image can be represented as a linear combination of several bases in a dictionary, thus producing a more concise representation. Through sparse representation, the basic structures of the image can be extracted, and the noise is represented by a small coefficients.

The noise in the sparse representation coefficients can be effectively reduced by convolutional denoising. In this step, convolutional kernels are employed to filter the sparse representation coefficients, eliminating noise components. Finally, through inverse sparse coding and inverse convolutional operations, the denoised coefficients are restored to the ultimate denoised image. Suppose that $Y_i \in \mathbb{R}^{h \times w}$ is the i-th training sample; the objective function can be written as follows:

$$\min_{A,\{X_i\}} \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2} \|A \otimes X_i - Y_i\|_2^2 + \lambda_X \psi(X_i) + \lambda_A \phi(A), \tag{5}$$

where $A \otimes X_i = \sum_{c=1}^{C} A_c * X_{i,c}$, $*$ is the two-dimensional convolutional operation, C is the number of channels, A represents the convolutional dictionary, $X_i = \{X_{i,c}\}_{c=1}^{C}$ is the sparse feature map of image Y_i, λ_{X_i} and λ_A are penalty parameters for X_i and A, $\psi(X_i)$ denotes the prior distribution of input variable X_i, and $\phi(A)$ is a regularization term for dictionary A.

Based on the learning method of convolutional dictionary mentioned above, Zheng et al. [33] proposed an improved deep convolutional dictionary learning (DCDicL) method. This method not only learns the deep prior of coefficient X but also learns the deep prior of convolutional dictionary A from the training data, and provides an adaptive dictionary for each image. The learning process can be expressed by the following formula:

$$\min_{\{A_i,X_i\}} \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2} \|A_i \otimes X_i - Y_i\|_2^2 + \lambda_X \psi(X_i) + \lambda_A \phi(A_i), \tag{6}$$

The expression above can be rewritten as the following optimization problem:

$$\begin{aligned} &\min \frac{1}{N} \sum_{i=1}^{N} F(A_i \otimes X_i, Y_i^{gt}) \\ &\text{s.t. } M\{A_i, X_i\} = \arg\min_{A,X} \frac{1}{2\sigma_i^2} \|A \otimes X - Y_i\|_2^2 + \lambda_X \psi(X) + \lambda_A \phi(A), \end{aligned} \tag{7}$$

where Y_i^{gt} is the ground truth version of Y_i, $F(,)$ is a function for measuring loss, and σ_i represents the level of noise.

DCDicL strictly follows the mathematical formula of dictionary learning, learning the prior knowledge of X and A from training data, and learning a specific dictionary

for each image. The dictionary can adapt to the image content and perceive the global information, thus giving DCDicL a powerful ability to recover the fine image structures even under severe noise. However, the DCDicL denoising method still cannot represent the fine structure of images very well, and its performance still needs to be improved.

3. Methodology

3.1. Proposed Method

Inspired by the related work, we put forward a deep convolutional dictionary learning denoising method based on distributed image patches (DCDicL+). This method not only processes the images globally but also processes the segmented image patches locally through similar patches. The proposed method enhances the restoration of fine details in the images. The fundamental method can be expressed as a minimization problem with the following formula:

$$\min_{A,X}\left\{\frac{1}{2}\|A\otimes X-Y\|_F^2+\lambda_1\psi(X)+\lambda_2\varphi(A)\right\}. \tag{8}$$

In this method, the complete noisy image is represented by Y, the global image dictionary is denoted by A, and X is introduced as the sparse coefficient.

3.2. Optimization Algorithm

Convolutional dictionary learning decomposes the entire image using convolutional operations, reducing redundancy in patch representations. Patch denoising helps preserve more details in the image. To obtain an effective method, Equation (8) is rewritten in the following forms of problems:

$$\begin{aligned}&\min_{\{A_l,X_l\}}\left\{\sum_{l=1}^{d}\frac{1}{2}\|A_l\otimes X_l-Y_l\|_F^2+\lambda_1\psi(X_l)+\lambda_2\phi(A_l)\right\},\\&\text{s.t. } A_l\otimes X_l\in M,\end{aligned} \tag{9}$$

where M is the given global constraint, λ_1 and λ_2 are penalty parameters for X_l and A_l, $\psi(X_l)$ denotes the prior distribution of input variable X_l, and $\phi(A_l)$ is a regularization term for dictionary A_l. By introducing auxiliary variables $X_l{}'$ and $A_l{}'$ to incorporate the denoising prior, Equation (9) can be expressed as

$$\begin{aligned}&\min_{X_l,X_l{}'}\left\{\frac{1}{2}\|A_l\otimes X_l{}'-Y_l\|_F^2+\lambda_1\psi(X_l)\right\},\\&\text{s.t. } X_l=X_l{}', A_l\otimes X_l\in M.\end{aligned} \tag{10}$$

$$\begin{aligned}&\min_{A_l,A_l{}'}\left\{\frac{1}{2}\|A_l{}'\otimes X_l-Y_l\|_F^2+\lambda_2\phi(A_l)\right\},\\&\text{s.t. } A_l=A_l{}', A_l\otimes X_l\in M.\end{aligned} \tag{11}$$

The constrained optimization problem above can be solved by using the half quadratic splitting (HQS) method [36]; then, the optimization problems are equivalent to minimizing the following two formulas:

$$\begin{aligned}&\min_{X_l,X_l{}'}\left\{\frac{1}{2}\|A_l\otimes X_l{}'-Y_l\|_F^2+\lambda_1\psi(X_l)+\frac{\mu_X}{2}\|X_l-X_l{}'\|_F^2\right\},\\&\text{s.t. } A_l\otimes X_l\in M.\end{aligned} \tag{12}$$

$$\begin{aligned}&\min_{A_l,A_l{}'}\left\{\frac{1}{2}\|A_l{}'\otimes X_l-Y_l\|_F^2+\lambda_2\phi(A_l)+\frac{\mu_A}{2}\|A_l-A_l{}'\|_F^2\right\},\\&\text{s.t. } , A_l\otimes X_l\in M,\end{aligned} \tag{13}$$

where μ_X and μ_A are penalty parameters. The above equations can be solved iteratively, and auxiliary variables ${X_l}^{\Theta}$ and ${A_l}^{\Theta}$ are introduced in the following formulas. For the update of sparse coefficient X_l, Equation (12) can be resolved by alternately optimizing the following subproblems:

$$\begin{aligned} {X_{l,(t)}}' &= \arg\min_{{X_l}^{\Theta}} \left\{ \frac{1}{2} \left\| A_{l,(t-1)} \otimes {X_l}^{\Theta} - Y_l \right\|_F^2 + \frac{\beta_X}{2} \left\| {X_l}^{\Theta} - X_{l,(t-1)} \right\|_F^2 \right\} \\ &= \frac{1}{\beta_X} F^{-1} \left\{ \left(a \circ y_{\uparrow C} + \beta_X x \right) - a \circ \left[\frac{\bar{a} \odot \left(a \circ y_{\uparrow C} + \beta_X x \right)}{\alpha_1 + (\bar{a} \odot a)} \uparrow_C \right] \right\}, \\ \text{s.t. } & \beta_X = \mu_X \sigma^2, y = F(Y_l), a = F(A_l), x = F(X_l), A_l \otimes X_l \in M. \end{aligned} \tag{14}$$

$$\begin{aligned} X_{l,(t)} &= \arg\min_{{X_l}^{\Theta}} \left\{ \psi({X_l}^{\Theta}) + \frac{\omega_X}{2} \left\| {X_{l,(t)}}' - {X_l}^{\Theta} \right\|_F^2 \right\} \\ &= prior({X_{l,(t-1)}}'), \\ \text{s.t. } & \omega_X = \mu_X / \lambda_1, A_l \otimes X_l \in M, \end{aligned} \tag{15}$$

where σ represents the noise level of the image patch and t denotes the number of iterations. Equation (14) obtains a closed-form solution for $X_{l,(t)}$ through fast Fourier transform (FFT) [37]. $F()$ and $F^{-1}()$ represent 2D FFT and its corresponding inverse transform, respectively. As a common operation in neural networks, the convolutional operation in the equations is typically implemented through matrix multiplication. $\circ$ is the Hadamard product. $\uparrow_C$ expands the channel dimension. $\bar{d} \odot d = \sum_{C=1}^{C} \bar{d}_C \circ d_C$. Equation (15) is updated through a neural network for the variables. $prior()$ represents the nonlinear mapping of the neural network. The prior knowledge of sparse coefficients X_l is learned through a U-Net structure [38]. For hyperparameters β_X and ω_X, we employ a configuration consisting of two Conv layers followed by a SoftPlus layer with input σ to predict the hyperparameters for each stage. For updating dictionary A_l, Equation (13) can be reformulated as the following subproblem:

$$\begin{aligned} {A_{l,(t)}}' &= \arg\min_{{A_l}^{\Theta}} \left\{ \frac{1}{2} \left\| {A_l}^{\Theta} \otimes X_{l,(t)} - Y_l \right\|_F^2 + \frac{\beta_A}{2} \left\| {A_l}^{\Theta} - A_{l,(t-1)} \right\|_F^2 \right\} \\ &= vec^{-1} \left\{ \left(\mathrm{X}^T \mathrm{X} + \beta_A I \right)^{-1} \left(\mathrm{X}^T \mathrm{Y} + \beta_A \mathrm{A} \right) \right\}, \\ \text{s.t. } & \beta_A = \mu_A \sigma^2, \mathrm{A} = vec^{-1}(A_l). \end{aligned} \tag{16}$$

$$\begin{aligned} A_{l,(t)} &= \arg\min_{{A_l}^{\Theta}} \left\{ \phi({A_l}^{\Theta}) + \frac{\omega_A}{2} \left\| {A_{l,(t)}}' - {A_l}^{\Theta} \right\|_F^2 \right\} \\ &= prior({A_{l,(t-1)}}'), \\ \text{s.t. } & \omega_A = \mu_A / \lambda_2, A_l \otimes X_l \in M, \end{aligned} \tag{17}$$

where $vec()$ is the vectorization operator and $vec^{-1}()$ reverses the vectorization. X, Y, and A denote the dimensional transformation results of matrices X_l, Y_l, and A_l, respectively. The convolutional multiplication in Equation (16) is expanded into matrix multiplication, and the solution for $A_{l,(t)}$ is obtained using the least squares method. Furthermore, Equation (17) is updated through a shallower neural network. Prior knowledge on dictionary A_l is learned through a relatively shallow network, consisting of six Conv layers with ReLU activation. For hyperparameters β_A and ω_A, we employ a configuration consisting of two Conv layers followed by a SoftPlus layer [39] with input σ to predict the hyperparameters for each stage. For updating global parameter M, by introducing auxiliary variables M' to incorporate the global denoising prior, the problem can be reformulated as the following formulas:

$${M_{(t)}}' = {M_{(t-1)}}' - \alpha_M \mu_M ({M_{(t-1)}}' - M_{(t-1)}). \tag{18}$$

$$M_{(t)} = M_{(t-1)} - \alpha_M \nabla f(M_{(t-1)}), \tag{19}$$

where α_M represents the learning rate and ∇ denotes the gradient. Specifically, $\nabla f(M_{(t)})$ represents the gradient of the objective function ($f(M)$) with respect to variable M at iteration t. The output image patch can be obtained through convolutional multiplication:

$$Y_{l,(t)}{}' = A_{l,(t)} \otimes X_{l,(t)} \tag{20}$$

3.3. Confidence-Weighted Fusion of Image Patches

The denoised image patches processed as mentioned above are combined into a complete output image according to specified offsets. Initially, a confidence matrix is generated by calculating the probability density values for each point in the image patches. Subsequently, the final denoised image is obtained by weighted summation of confidence-weighted input image patches. This approach ensures the preservation of high-confidence information and can be expressed as

$$P\left(Y_{l,(t)}{}'\right) = \frac{\exp\left(-\frac{1}{2}\left(Y_{l,(t)}{}' - \alpha\right)^T \Sigma^{-1}\left(Y_{l,(t)}{}' - \alpha\right) + \tau M\right)}{\sqrt{(2\pi)^k \det \Sigma}}, \tag{21}$$

$$Y' = \left\{ \begin{array}{l} \sum\limits_t \left[P\left(Y_{1,(t)}{}'\right) \times Y_{1,(t)}{}'\right], \\ \sum\limits_t \left[P\left(Y_{2,(t)}{}'\right) \times Y_{2,(t)}{}'\right], \\ \ldots, \sum\limits_t \left[P\left(Y_{d,(t)}{}'\right) \times Y_{d,(t)}{}'\right] \end{array} \right\}, \tag{22}$$

where $P()$ denotes the probability density function of a multivariate Gaussian distribution, $Y_l{}'$ is the image patch processed after denoising, Y' is the fully assembled denoised image, τ represents the penalty term for the global component, $\sum$ represents the covariance matrix, $\det \sum$ is the determinant of $\sum$, k is the dimensionality, and α denotes the mean.

In conclusion, the solution steps of the method proposed in this paper are shown in Algorithm 1. More specifically, InitNet, consisting of two Conv layers with ReLU activation, is employed to learn initialization coefficient $X_{l,(0)}$. Subsequently, for each of the t iterations, the value of $X_{l,(t)}{}'$ is calculated by Equation (14). This result is then fed into Equation (15), which includes three connected up-sampling blocks and three down-sampling blocks, allowing for the estimation of the sparse coefficient by using the prior learned from the training data. Similar procedures are applied for updating dictionary $A_{l,(t)}$ and $M_{(t)}$. The final image is obtained through convolutional operation and confidence fusion.

The effectiveness of our proposed method will be analyzed in the experimental part of Section 4.

Algorithm 1: DCDicL+

Input: Noisy Image Y, dictionary D ;
Initialize: Y_l, $A_{l,(0)} = 0$, $X_{l,(0)} = InitNet\left(Y_{l,(0)}, \sigma\right)$, $t = 0$;
for $t \leftarrow 1$ **to** T **do**
  compute ${X_{l,(t)}}'$, via Equation (14);
  compute $X_{l,(t)}$, via Equation (15);
  compute ${A_{l,(t)}}'$, via Equation (16);
  compute $A_{l,(t)}$, via Equation (17);
  compute ${M_{(t)}}'$, via Equation (18);
  compute $M_{(t)}$, via Equation (19);
  compute ${Y_{l,(t)}}'$, via Equation (20);
  compute $P\left({Y_{l,(t)}}'\right)$, via Equation (21);
  $t = t + 1$
Output: Y';

4. Experimental Results

In this section, we report the experimental results. The training data were generated by using the WED [40] (4744 images), DIV2K [41] (900 images), and BSD400 [42] (400 images) datasets. The tests were conducted on the grayscale datasets Set12 (12 images), BSD68 [43] (68 images), and Urban100 [44] (100 images) and the color datasets Urban100 [44] (100 images), CBSD48 [43] (48 images), and Kodak24 [45] (24 images). All experiments were carried out on the GTX 1080Ti GPU with CUDA version 11.2 using PyTorch.

Peak signal-to-noise ratio (PSNR) [46] and structural similarity (SSIM) [47] were employed for assessing the local consistency within image patches. Additionally, the consistency between different local regions was evaluated with the mean square error (MSE) [48]. For the given clean image Y' and noisy image Y with a size of $m \times n$, MSE is defined as follows:

$$MSE = \frac{1}{\mathrm{mn}} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \left[Y'(i,j) - Y(i,j)\right]^2,$$

where the PSNR value is contingent on the MSE, implying that a smaller MSE corresponds to a higher PSNR value. This relationship indicates a diminished distinction between the reconstructed image and the actual image. PSNR is defined as

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX^2}{MSE}\right),$$

where MAX represents the maximum pixel value of the image. In parallel, SSIM serves as a metric for gauging the likeness between the reconstructed and actual images in dimensions such as brightness, contrast, and structure. SSIM can be mathematically formulated as follows:

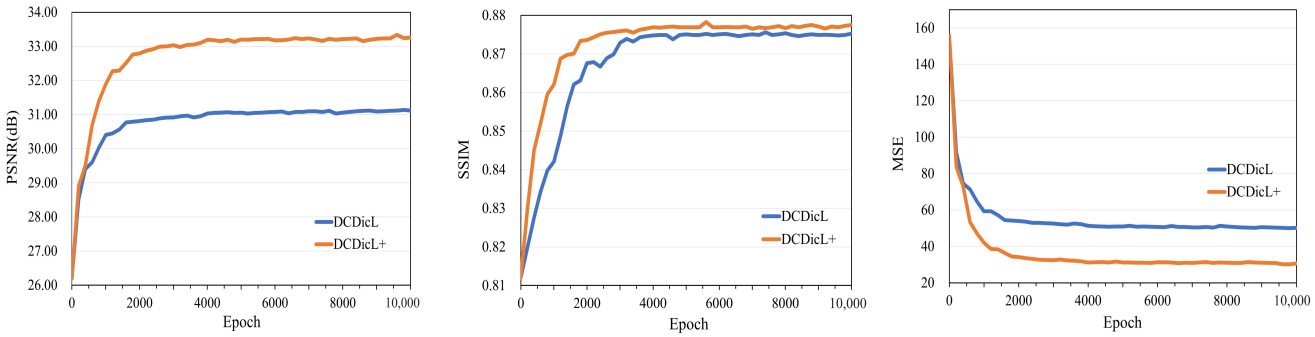
$$SSIM = \frac{(2\mu_{Y'}\mu_Y + c_1)(2\sigma_{Y'Y} + c_2)}{(\mu_{Y'}^2 + \mu_Y^2 + c_1)(\varepsilon_{Y'}^2 + \varepsilon_Y^2 + c_2)},$$

where $\sigma_{Y'Y}$ is the covariance between the reconstructed and original images, $\mu_{Y'}$ and μ_Y represent the means, $\varepsilon_{Y'}$ and ε_Y represent the variance, and c_1 and c_2 are constants. A greater SSIM value denotes heightened similarity between the two images, signifying superior image quality.

4.1. Parameter Settings

4.1.1. Training Iterations

For each input image, the image patch size was set to 128×128, and the confidence merging step size was set to 8. During the training process, Gaussian noise with zero average value and a noise level in the range of $\sigma \in [0, 50]$ was added to the images. These noisy images were trained in batches. The learning rate was set to 10^{-4}, and the batch size was set to 16 on the GTX 1080Ti GPU. The loss function utilized L_1 loss, and the Adam optimizer [49] was employed for updating the network parameters. Iteration T was carried out in four stages, which are discussed in Section 4.3. The number of atoms in dictionary A was fixed at 64, and the size of A was set to 5. Simultaneously, to ensure the convergence of the training function, this study employed a self-constructed dataset for validation. Figure 2a,b illustrate the changes in PSNR (dB) and SSIM with the increase in training iterations when the noise level is $\sigma = 25$. Figure 2c illustrates the changes in MSE with the increase in training iterations. It can be observed that with a greater number of iterations, the values of PSNR and SSIM gradually improve, and MSE gradually converges. After 4000 iterations, a favorable outcome is achieved, prompting the decision to establish the final number of iterations at 4000.



(**a**) (**b**) (**c**)

Figure 2. Contrast curves of noise level of $\sigma = 25$ during training. (**a**) PSNR (dB) trends of DCDicL and DCDicL+, (**b**) SSIM trends of DCDicL and DCDicL+, and (**c**) MSE trends of DCDicL and DCDicL+.

4.1.2. Confidence Scaling Factor

Different values of confidence scaling factor p produce different regions of interest, potentially resulting in excessive smoothing or loss of local detail information in the image. Parameter p controls the size of the covariance matrix for confidence-weighted multivariate normal distributions, influencing the spatial range of the multivariate normal distribution around the center of the image patch and consequently affecting the confidence distribution of each pixel. A larger p value leads to a wider confidence distribution, exerting a greater impact on surrounding pixels during the mosaic process, whereas a smaller p value results in a more localized confidence distribution.

To assess the influence of different confidence scaling factors, this study additionally performed ablation analysis by varying the scaling factor (p), demonstrating that $p = 2$ is the optimal parameter selection. Moreover, it was observed that further increases in p did not significantly enhance performance. Further details can be found in Section 4.3. The experimental results are shown in Tables 1 and 2 (where the bold values represent the best values).

Compared with $p = 0.5$ for original input images with noise levels of $\sigma = 15, 25$, and 50 in the grayscale dataset Urban100, using $p = 2$ for denoising can increase the PSNR by 0.01 dB, 0.02 dB, and 0.03 dB and SSIM by 0.0001, 0.0003, and 0.0006, respectively. In the color dataset Urban100, compared with $p = 0.5$, using $p = 2$ for denoising can increase the PSNR by 0.01 dB, 0.02 dB, and 0.02 dB and SSIM by 0.0002, 0.0002, and 0.0003, respectively.

The experiments indicate that when the scale factor (p) is set to 2, the values of the PSNR and SSIM are better, so $p = 2$ was set in subsequent experiments.

Table 1. Comparison of PSNR (dB)/SSIM average values with different scaling factors on grayscale datasets.

Dataset	Noise	$p = 0.5$	$p = 2$
Set12	15	34.45/0.9142	34.45/**0.9143**
	25	33.17/0.8765	**33.19**/**0.8769**
	50	31.25/0.8109	**31.87**/**0.8116**
BSD68	15	32.14/0.9136	32.14/**0.9137**
	25	31.15/0.8577	**31.16**/**0.8579**
	50	30.23/0.7292	**30.25**/**0.7295**
Urban100	15	33.42/0.9472	**33.43**/**0.9473**
	25	32.20/0.9179	**32.22**/**0.9182**
	50	30.93/0.8556	**30.96**/**0.8562**

Table 2. Comparison of PSNR (dB)/SSIM average values with different scaling factors on color datasets.

Dataset	Noise	$p = 0.5$	$p = 2$
Set12	15	34.45/0.9142	34.45/**0.9143**
	25	34.16/0.9460	**34.18**/**0.9462**
	50	32.46/0.9014	**32.48**/**0.9017**
CBSD68	15	34.90/0.9381	**34.91**/0.9381
	25	33.32/0.8976	**33.34**/**0.8977**
	50	31.94/0.8156	**31.96**/**0.8159**
Kodak24	15	35.70/0.9316	**35.71**/**0.9317**
	25	34.11/0.8947	**34.13**/**0.8949**
	50	32.63/0.8230	**32.64**/**0.8233**

4.2. Comparison Results

In order to verify the proposed denoising method, DCDicL+, noise levels of 15, 25, and 50 were added to the original input images. DCDicL+ was compared with several representative denoising methods, including BM3D [20], WNNM [23], FFDNet [27], DnCNN [18], SwinIR [45], LKSVD [31], DCDicL [33], and SCUNet [47]. The implementation codes of these comparison methods are provided by their respective authors (https://github.com/natezhenghy/DCDicL_denoising, accessed on 17 December 2023) and tested on the same datasets.

4.2.1. Grayscale Datasets

As shown in Table 3, the best PSNR and SSIM for each noise level in each dataset are displayed in bold, indicating the best results, and those underlined indicate the second best results. It can be observed that compared with other denoising methods, the proposed denoising method achieves superior performance on the grayscale datasets Set12, BSD68, and Urban100. The improvement range for the PSNR is 0.15 dB to 11.03 dB, and the improvement range for SSIM is 0.0002 to 0.1851. On the dataset Set12, compared with DCDicL, the PSNR/SSIM of the proposed method are increased by 1.11 dB/0.0028, 2.16 dB/0.0021, and 3.87 dB/0.0012, respectively; Compared with SCUNet, they are increased by 1.02 dB/0.0126, 2.10 dB/0.0015, and 3.83 dB/0.0002, respectively. This can be attributed to the excellent learning capability of DCDicL+ with a large number of training data. The proposed method explores the inherent characteristics in images and performs finer denoising and restoration of the fine structure in the images.

Table 3. Comparison of PSNR (dB)/SSIM average values on different grayscale datasets.

Dataset / Method	Set12			BSD68			Urban100		
	$\sigma = 15$	$\sigma = 25$	$\sigma = 50$	$\sigma = 15$	$\sigma = 25$	$\sigma = 50$	$\sigma = 15$	$\sigma = 25$	$\sigma = 50$
BM3D	32.37/ 0.8952	29.97/ 0.8504	26.72/ 0.7676	31.07/ 0.8717	28.57/ 0.8013	25.60/ 0.6864	32.35/ 0.9220	29.70/ 0.8777	25.95/ 0.7791
WNNM	32.71/ 0.8988	30.26/ 0.8557	27.05/ 0.7775	31.37/ 0.8766	28.83/ 0.8087	25.87/ 0.6982	32.53/ 0.9271	30.38/ 0.8885	26.83/ 0.8047
FFDNet	32.75/ 0.9024	30.43/ 0.8631	27.32/ 0.7899	31.63/ 0.8902	29.19/ 0.8288	26.29/ 0.7239	32.40/ 0.9265	29.90/ 0.8979	26.50/ 0.8048
DnCNN	32.86/ 0.9024	30.44/ 0.8617	27.18/ 0.7828	31.73/ 0.8907	29.23/ 0.8279	26.23/ 0.7189	32.64/ 0.9246	29.95/ 0.8781	26.26/ 0.7856
SwinIR	33.36/ 0.8898	31.01/ 0.8482	27.91/ 0.8119	31.97/ 0.8959	29.50/ 0.8321	26.58/ 0.7298	32.70/ 0.9104	31.30/ 0.8724	27.98/ 0.8039
LKSVD	28.72/ 0.7826	24.85/ 0.7354	20.84/0 0.7042	28.48/ 0.7835	24.96/ 0.7371	20.97/ 0.7035	28.46/ 0.7836	23.80/ 0.7331	20.22/ 0.7018
DCDicL	33.34/ 0.9115	31.03/ 0.8748	28.00/ 0.8122	31.95/ 0.8957	29.52/ 0.8379	26.63/ 0.7395	32.31/ 0.9383	29.65/ 0.9129	26.22/ 0.8336
SCUNet	33.43/ 0.9017	31.09/ 0.8754	28.04/ 0.8132	31.99/ 0.9001	29.55/ 0.8402	26.67/ 0.7456	32.88/ 0.9458	31.58/ 31.58/	28.56/ 0.8392
Our	**34.45/ 0.9143**	**33.19/ 0.8769**	**31.87/ 0.8134**	**32.14/ 0.9137**	**31.16/ 0.8579**	**30.25/ 0.7478**	**33.43/ 0.9473**	**32.22/ 0.9182**	**30.96/ 0.8562**

Simultaneously, this paper selected two images from Set12 and introduced Gaussian noise at the level of $\sigma = 50$ to further validate the visual denoising effect. The experimental results are shown in Figures 3 and 4. The green boxes are enlarged images of the red boxes. Upon closer inspection of the enlarged comparison diagrams, it becomes evident that DCDicL+ restores more edge and texture information. According to the analysis of specific numerical values, the proposed method demonstrates significant improvement compared with other methods on the image Starfish, exhibiting enhancements ranging from 0.98 dB to 6.92 dB in the PSNR and improvements from 0.0207 to 0.1504 in SSIM. Likewise, on the image House, the proposed method exhibits clear superiority over other methods, manifesting improvements in the PSNR from 2.57 dB to 12.59 dB and enhancements in SSIM from 0.0308 to 0.1007.

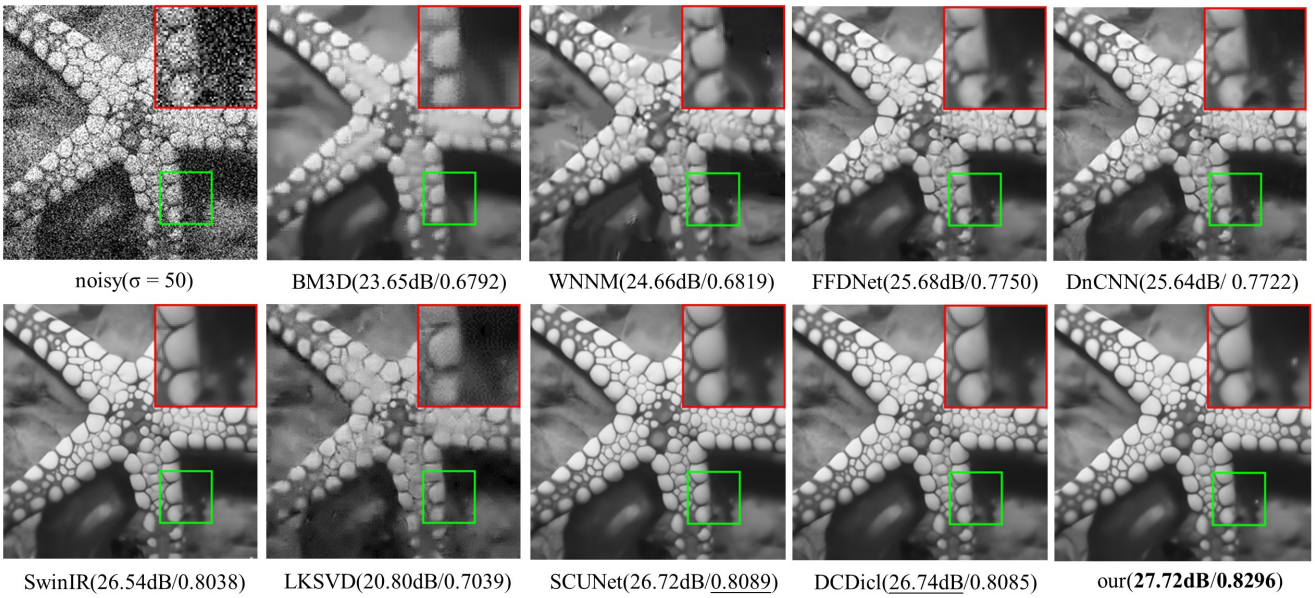


Figure 3. Grayscale image Starfish with noise level $\sigma = 50$: results of different methods (PSNR (dB)/SSIM).

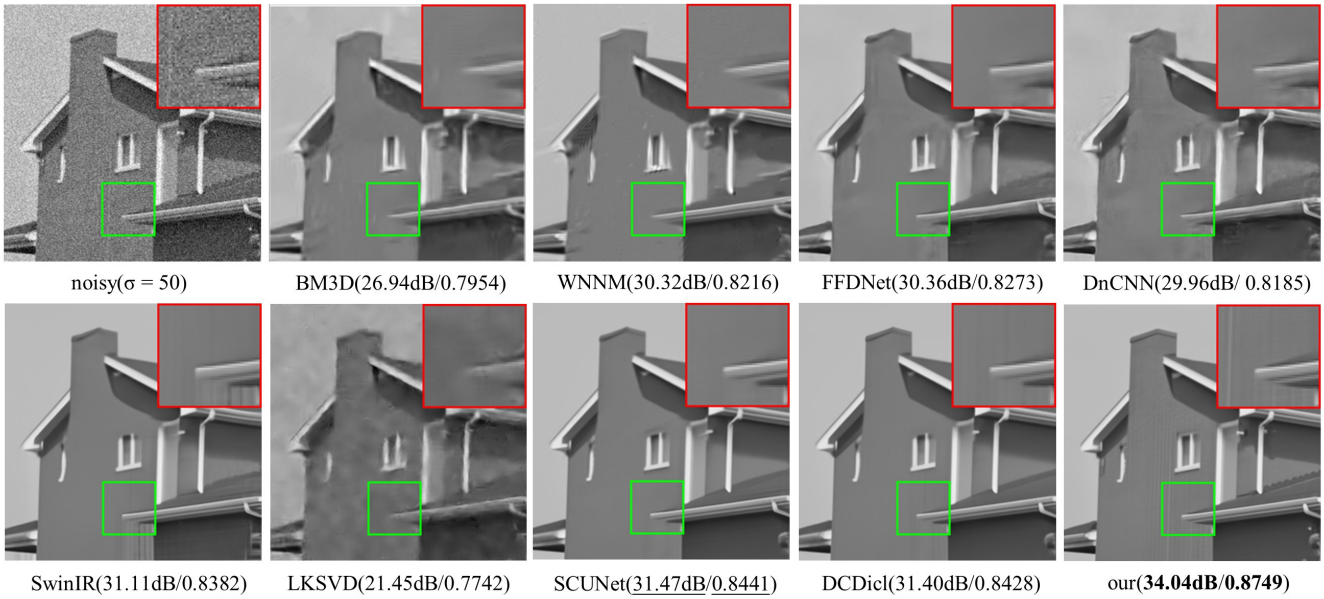


Figure 4. Grayscale image House with noise level $\sigma = 50$: results of different methods (PSNR (dB)/SSIM).

4.2.2. Color Datasets

For the color datasets Urban100, CBSD68, and Kodak24, Table 4 provides a comparative analysis of various methods based on the PSNR and SSIM. Compared with DCDicL, the PSNR/SSIM of the proposed method are improved by 0.91 dB/0.0149, 1.41 dB/0.0162, and 3.49 dB/0.0133, respectively, on the color dataset Urban100. Compared with SCUNet, these performance metrics are improved by 0.63 dB/0.0107, 1.15 dB/0.0120, and 2.34 dB/0.0116, respectively. It can be observed that the proposed method also demonstrates effective denoising performance on color images and restores them effectively.

Table 4. Comparison of PSNR (dB)/SSIM average values on different color datasets.

Dataset / Method	Urban100			CBSD68			Kodak24		
	$\sigma = 15$	$\sigma = 25$	$\sigma = 50$	$\sigma = 15$	$\sigma = 25$	$\sigma = 50$	$\sigma = 15$	$\sigma = 25$	$\sigma = 50$
BM3D	33.93/ 0.9408	31.36/ 0.9092	27.93/ 27.93/	33.50/ 0.9215	30.69/ 0.8672	27.36/ 0.7626	34.26/ 0.9147	31.67/ 0.8670	28.44/ 0.7760
WNNM	32.86/ 0.9024	32.86/ 0.9024	27.18/ 0.7828	31.73/ 0.8907	31.73/ 0.8907	26.23/ 0.7189	32.64/ 0.9246	29.95/ 0.8781	26.26/ 0.7856
FFDNet	33.83/ 0.9418	31.40/ 0.9120	28.05/ 0.8476	33.87/ 0.9290	31.21/ 0.8821	27.96/ 0.7887	34.63/ 0.9224	32.13/ 0.8791	28.98/ 0.7952
DnCNN	32.98/ 0.9314	30.81/ 0.9015	27.59/ 0.8331	33.89/ 0.9290	31.23/ 0.8830	27.92/ 0.7896	34.48/ 0.9209	32.03/ 0.8775	28.85/ 0.7917
SwinIR	35.13/ 0.9532	32.90/ 0.9284	29.82/ 0.8675	34.42/ 0.9203	31.78/ 0.8862	28.56/ 0.7890	35.34/ 0.9212	32.89/ 0.8864	29.79/ 0.8041
LKSVD	32.76/ 0.8106	30.37/ 0.7798	27.12/ 0.7187	31.63/ 0.8282	29.15/ 0.7848	26.19/ 0.7169	32.46/ 0.8236	29.80/ 0.7831	26.22/ 0.7318
DCDicL	34.90/ 0.9511	32.77/ 0.9300	28.99/ 0.8884	34.36/ 0.9348	31.75/ 0.8930	28.57/ 0.8107	35.38/ 0.9300	32.97/ 0.8928	29.96/ 0.8219
SCUNet	35.18/ 0.9553	33.03/ 0.9342	30.14/ 0.8901	34.40/ 0.9351	31.79/ 0.8931	28.61/ 0.8130	35.34/ 0.9293	32.92/ 0.8922	29.87/ 0.8198
Our	**35.81/ 0.9660**	**34.18/ 0.9462**	**32.48/ 0.9017**	**34.91/ 0.9381**	**33.34/ 0.8977**	**31.96/ 0.8159**	**35.71/ 0.9317**	**34.13/ 0.8949**	**32.64/ 0.8233**

Similarly, from the visual representation in Figures 5 and 6, it can be observed that the proposed method exhibits the best visual performance. As shown in Figure 5, compared with other methods, the PSNR and SSIM of this method are improved by 0.38 dB to

5.81 dB and by 0.0021 to 0.1783, respectively. As shown in Figure 6, it can be observed that compared with other methods, DCDicL+ can recover a finer structure, achieving an enhancement of 0.11 dB to 6.00 dB in the PSNR and an improvement of 0.0032 to 0.1879 in SSIM. This fully proves the good performance of the denoising method proposed in this paper.

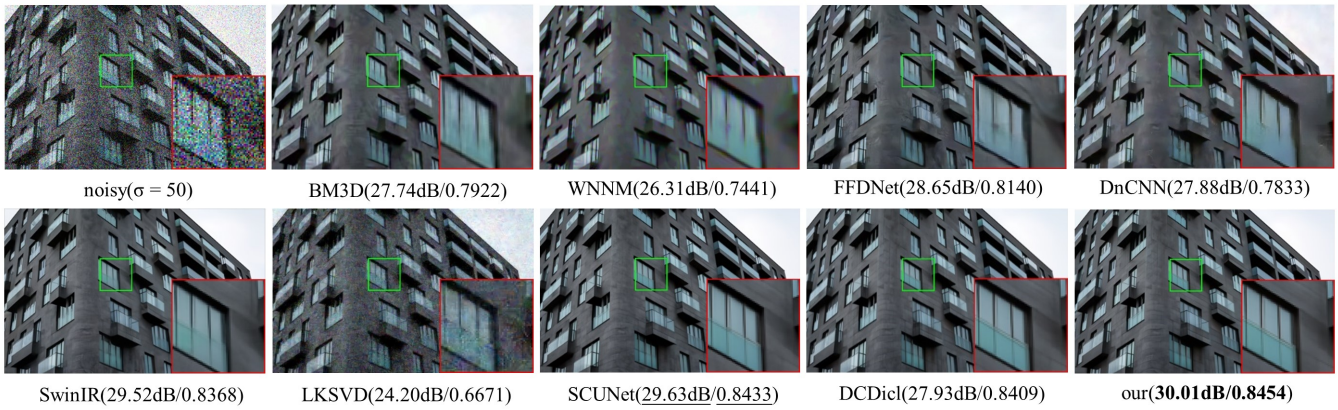


Figure 5. Color image with noise level $\sigma = 50$ denoised with different methods (PSNR (dB) /SSIM).

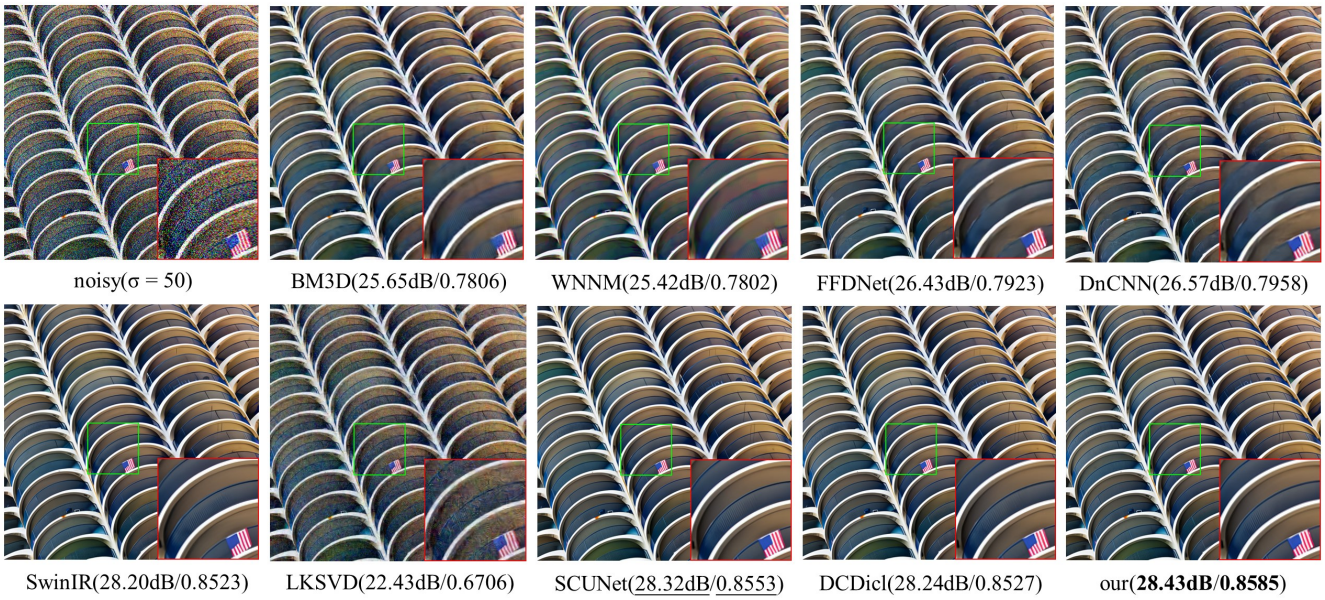
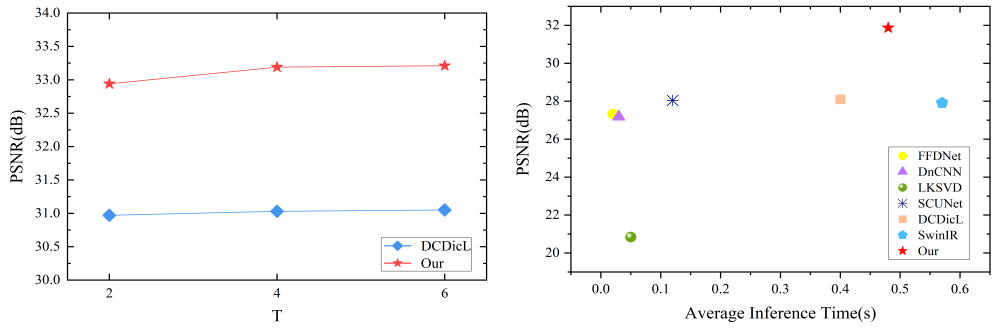


Figure 6. Color image with noise level $\sigma = 50$ denoised with different methods (PSNR (dB) /SSIM).

4.3. Discussion

Figure 7 shows the experimental results of DCDicL+ compared with other methods. All experiments were conducted on the dataset Set12. Figure 7a shows the PSNR comparison between the DCDicL+ and DCDicL methods in different stages T; the experiment was carried out at $\sigma = 25$. It can be seen that the PSNR continues to increase with the increase in stage T, but this also leads to an increase in time. However, we can also see that the increase in PSNR is very small after setting stage T to 4, so we choose the value of stage T as 4. Under the same conditions, the PSNR value of the proposed method is about 2 dB higher than that of the DCDicL method. Furthermore, the inference time of the proposed method was compared with that of the competitive methods; all experiments were carried out at $\sigma = 50$. As can be observed in Figure 7b, DCDicL+ is slower than DnCNN, FFDNet, LKSVD, and SCUNet, but it achieves better output in terms of PSNR. Compared with SwinIR, the proposed method demonstrates higher performance in terms of PSNR and requires less inference time, further confirming its effectiveness. In conclusion,

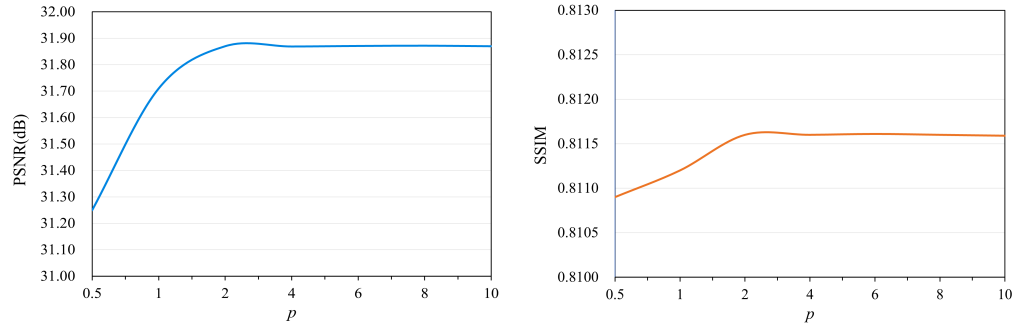
it can be deduced that DCDicL+ provides a robust solution in terms of both effectiveness and efficiency.



(**a**) (**b**)

Figure 7. Performance comparison of DCDicL+ with other methods. (**a**) Comparison of PSNR (dB) obtained with DCDicL at different stages T. (**b**) Comparison of inference time and PSNR (dB) of different methods.

For the experimental analysis, we systematically varied the p values from 0.5 to 10 to comprehensively evaluate their impact on denoising performance. The experiments were conducted using the dataset Set12 with a noise level of $\sigma = 50$. The results, shown in Figure 8, reveal a notable trend: both the PSNR (dB) and SSIM exhibit a discernible pattern of convergence when $p = 2$. This convergence indicates that the denoising performance reaches a stable state at this specific p value. Given this observation, we deduce that further adjustments beyond $p = 2$ do not result in significant improvements in denoising quality. Consequently, we conclude that $p = 2$ represents the optimal parameter selection for our denoising method in terms of achieving the desired balance between edge preservation and noise reduction.



(**a**) (**b**)

Figure 8. Variation curves of different p at noise level $\sigma = 50$. (**a**) PSNR (dB) trend of DCDicL and (**b**) SSIM trend of DCDicL.

To evaluate the efficacy of our method in preserving edge details and overall image quality, we utilized the Edge Preservation Index (EPI) [50], a quantitative measure of edge sharpness and naturalness in denoised images. A higher EPI value, closer to 1, indicates better edge preservation. The experimental images were obtained from the dataset Set12. Table 5 illustrates the comparison between our method and DCDicL in preserving edge details across various images. Notably, our method demonstrates better performance. For instance, at noise levels of $\sigma = 15, 25$, and 50, the EPI values for the image Peppers are 0.0103, 0.0073, and 0.0030 higher, respectively, compared with DCDicL. Furthermore, the average EPI values are higher than those of DCDicL by 0.0027, 0.0030, and 0.0051, respectively.

Table 5. Comparison of EPI obtained with DCDicL on Set12.

Method / Image	$\sigma = 15$		$\sigma = 25$		$\sigma = 50$	
	DCDicL	**Our**	**DCDicL**	**Our**	**DCDicL**	**DCDicL**
C.man	**0.7705**	0.7702	**0.6929**	0.6903	**0.5845**	0.5840
Peppers	0.7445	**0.7548**	0.5765	**0.5838**	0.4968	**0.4998**
House	0.7633	0.7633	**0.7204**	0.7194	0.6513	**0.6515**
Airplane	0.8480	**0.8482**	0.7996	**0.8011**	0.7155	**0.7193**
Couple	0.8685	**0.8720**	0.8312	**0.8369**	0.7542	**0.7651**
Parrot	0.7598	**0.7616**	0.6870	**0.6904**	0.5749	**0.5797**
Man	0.8144	**0.8167**	0.7447	**0.7485**	0.6574	**0.6623**
Monarch	0.6675	**0.6701**	0.6067	**0.6116**	0.5136	**0.5171**
Starfish	0.8743	**0.8815**	0.8395	**0.8500**	0.7533	**0.7804**
Boat	0.6598	**0.6642**	0.5865	**0.5888**	0.4796	**0.4824**
Barbara	**0.7205**	0.7203	0.6261	**0.6266**	0.4789	**0.4826**
Lena	0.7209	**0.7213**	**0.6521**	0.6513	**0.5442**	0.5422
Average	0.7677	**0.7704**	0.6969	**0.6999**	0.6004	**0.6055**

5. Conclusions

In order to solve the noise interference problem of Micro-LED displays, a deep convolutional dictionary learning denoising method based on distributed image patches is proposed. The initial stage includes preprocessing the original image into locally distributed image patches with uniform size and then obtaining the non-local self-similar sparse representation dictionary customized for these patches. Subsequently, the introduction of deep convolutional neural networks promotes global self-similarity matching, and under the guidance of a confidence evaluation function, weighted fusion is used to denoise the image. Notably, the experimental results show that the method is superior in preserving the details of complex images, thus reducing the potential loss. In rigorous objective and subjective evaluations, the proposed method consistently outperforms all comparative methods, establishing itself as a novel and effective denoising solution for Micro-LED displays.

In future works, there is potential for enhancing the adaptability of the denoising method to diverse noise profiles and addressing more complex denoising challenges. Additionally, exploring its scalability and applicability in Micro-LED display technologies will be crucial. Integration with other image processing techniques could further improve overall performance and contribute to advancements in image quality enhancement.

Author Contributions: All authors contributed to this study, and the specific contributions of each author are as follows: L.Y., W.G., and J.L. completed the methodology, collected the experimental data, and wrote the manuscript. J.L. and W.G. completed the revision and checking of the manuscript. L.Y. and J.L. provided financial support and supervised this work. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the National Natural Science Foundation of China under grant 62204044, in part by the State Key Laboratory of Integrated Chips and Systems under grant SKLICS-K202302, and in part by the Science and Technology Commission of Shanghai Municipality Program under grants 20010500100 and 21511101302.

Data Availability Statement: The data that support the findings of this study are available online. Data download web address: https://github.com/wayoungg/Deep-Convolutional-Dictionary-Learning-Denoising-Method-Based-on-Distributed-Image-Patches. (accessed on 15 March 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Anwar, A.R.; Sajjad, M.T.; Johar, M.A.; Hernandez-Gutierrez, C.A.; Usman, M.; Lepkowski, S. Recent progress in micro-LED-based display technologies. *Laser Photonics Rev.* **2022**, *16*, 2100427.
2. Lin, C.C.; Wu, Y.R.; Kuo, H.C.; Wong, M.S.; DenBaars, S.P.; Nakamura, S.; Pandey, A.; Mi, Z.; Tian, P.; Ohkawa, K.; et al. The micro-LED roadmap: Status quo and prospects. *J. Phys. Photonics* **2023**, *5*, 042502.

3. Chen, D.; Chen, Y.C.; Zeng, G.; Zhang, D.W.; Lu, H.L. Integration Technology of Micro-LED for Next-Generation Display. *Research* **2023**, *6*, 0047.
4. Pandey, A.; Min, J.; Reddeppa, M.; Malhotra, Y.; Xiao, Y.; Wu, Y.; Sun, K.; Mi, Z. An Ultrahigh Efficiency Excitonic Micro-LED. *Nano Lett.* **2023**, *23*, 1680–1687.
5. Zhu, G.; Liu, Y.; Ming, R.; Shi, F.; Cheng, M. Mass transfer, detection and repair technologies in micro-LED displays. *Sci. China Mater.* **2022**, *65*, 2128–2153.
6. Lin, J.; Jiang, H. Development of microLED. *Appl. Phys. Lett.* **2020**, *116*, 100502.
7. Chen, Z.; Yan, S.; Danesh, C. MicroLED technologies and applications: Characteristics, fabrication, progress, and challenges. *J. Phys. D Appl. Phys.* **2021**, *54*, 123001.
8. James Singh, K.; Huang, Y.M.; Ahmed, T.; Liu, A.C.; Huang Chen, S.W.; Liou, F.J.; Wu, T.; Lin, C.C.; Chow, C.W.; Lin, G.R.; et al. Micro-LED as a promising candidate for high-speed visible light communication. *Appl. Sci.* **2020**, *10*, 7384.
9. Hsiang, E.L.; Yang, Z.; Yang, Q.; Lan, Y.F.; Wu, S.T. Prospects and challenges of mini-LED, OLED, and micro-LED displays. *J. Soc. Inf. Disp.* **2021**, *29*, 446–465.
10. Zhang, X.; Yin, L.; Ren, K.; Zhang, J. Research on Simulation Design of MOS Driver for Micro-LED. *Electronics* **2022**, *11*, 2044.
11. Mohammed Abd-Alsalam Selami, A.; Freidoon Fadhil, A. A study of the effects of gaussian noise on image features. *Kirkuk Univ. J.-Sci. Stud.* **2016**, *11*, 152–169.
12. Fan, L.; Zhang, F.; Fan, H.; Zhang, C. Brief review of image denoising techniques. *Vis. Comput. Ind. Biomed. Art* **2019**, *2*, 7.
13. Buades, A.; Coll, B.; Morel, J.M. A review of image denoising algorithms, with a new one. *Multiscale Model. Simul.* **2005**, *4*, 490–530.
14. Tian, C.; Fei, L.; Zheng, W.; Xu, Y.; Zuo, W.; Lin, C.W. Deep learning on image denoising: An overview. *Neural Netw.* **2020**, *131*, 251–275.
15. Goyal, B.; Dogra, A.; Agrawal, S.; Sohi, B.S.; Sharma, A. Image denoising review: From classical to state-of-the-art approaches. *Inf. Fusion* **2020**, *55*, 220–244.
16. Yu, S.; Ma, J.; Wang, W. Deep learning for denoising. *Geophysics* **2019**, *84*, V333–V350.
17. Tomasi, C.; Manduchi, R. Bilateral filtering for gray and color images. In Proceedings of the Sixth International Conference on Computer Vision, Bombay, India, 4–7 January 1998; IEEE Cat. No. 98CH36271; IEEE: New York, NY, USA, 1998; pp. 839–846.
18. Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; Zhang, L. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Trans. Image Process.* **2017**, *26*, 3142–3155.
19. Srivastava, M.; Anderson, C.L.; Freed, J.H. A new wavelet denoising method for selecting decomposition levels and noise thresholds. *IEEE Access* **2016**, *4*, 3862–3877.
20. Dabov, K.; Foi, A.; Katkovnik, V.; Egiazarian, K. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Trans. Image Process.* **2007**, *16*, 2080–2095.
21. Aharon, M.; Elad, M.; Bruckstein, A. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Process.* **2006**, *54*, 4311–4322.
22. Jalalzai, K. Some remarks on the staircasing phenomenon in total variation-based image denoising. *J. Math. Imaging Vis.* **2016**, *54*, 256–268.
23. Gu, S.; Zhang, L.; Zuo, W.; Feng, X. Weighted nuclear norm minimization with application to image denoising. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 2862–2869.
24. Yao, C.; Jin, S.; Liu, M.; Ban, X. Dense residual Transformer for image denoising. *Electronics* **2022**, *11*, 418.
25. Tian, C.; Xu, Y.; Fei, L.; Yan, K. Deep learning for image denoising: A survey. In Proceedings of the Genetic and Evolutionary Computing: Proceedings of the Twelfth International Conference on Genetic and Evolutionary Computing, Changzhou, China, 14–17 December 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 563–572.
26. Zhang, K.; Zuo, W.; Gu, S.; Zhang, L. Learning deep CNN denoiser prior for image restoration. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3929–3938.
27. Zhang, K.; Zuo, W.; Zhang, L. FFDNet: Toward a fast and flexible solution for CNN-based image denoising. *IEEE Trans. Image Process.* **2018**, *27*, 4608–4622.
28. Alsaiari, A.; Rustagi, R.; Thomas, M.M.; Forbes, A.G.; Alhakamy, A. Image denoising using a generative adversarial network. In Proceedings of the 2019 IEEE 2nd International Conference on Information and Computer Technologies (ICICT), Kahului, HI, USA, 14–17 March 2019; IEEE: New York, NY, USA, 2019; pp. 126–132.
29. Im Im, D.; Ahn, S.; Memisevic, R.; Bengio, Y. Denoising criterion for variational auto-encoding framework. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31.
30. Simon, D.; Elad, M. Rethinking the CSC model for natural images. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 11.
31. Scetbon, M.; Elad, M.; Milanfar, P. Deep k-svd denoising. *IEEE Trans. Image Process.* **2021**, *30*, 5944–5955.
32. Daubechies, I.; Defrise, M.; De Mol, C. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun. Pure Appl. Math. A J. Issued Courant Inst. Math. Sci.* **2004**, *57*, 1413–1457.
33. Zheng, H.; Yong, H.; Zhang, L. Deep convolutional dictionary learning for image denoising. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 630–641.
34. Bian, S.; He, X.; Xu, Z.; Zhang, L. Hybrid Dilated Convolution with Attention Mechanisms for Image Denoising. *Electronics* **2023**, *12*, 3770.

35. Ilesanmi, A.E.; Ilesanmi, T.O. Methods for image denoising using convolutional neural network: A review. *Complex Intell. Syst.* **2021**, *7*, 2179–2198.
36. Sun, Y.; Yang, Y.; Liu, Q.; Chen, J.; Yuan, X.T.; Guo, G. Learning non-locally regularized compressed sensing network with half-quadratic splitting. *IEEE Trans. Multimed.* **2020**, *22*, 3236–3248.
37. Reddy, B.S.; Chatterji, B.N. An FFT-based technique for translation, rotation, and scale-invariant image registration. *IEEE Trans. Image Process.* **1996**, *5*, 1266–1271.
38. Huang, H.; Lin, L.; Tong, R.; Hu, H.; Zhang, Q.; Iwamoto, Y.; Han, X.; Chen, Y.W.; Wu, J. Unet 3+: A full-scale connected unet for medical image segmentation. In Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; IEEE:New York, NY, USA, 2020; pp. 1055–1059.
39. Zheng, H.; Yang, Z.; Liu, W.; Liang, J.; Li, Y. Improving deep neural networks using softplus units. In Proceedings of the 2015 International Joint Conference on Neural Networks (IJCNN), Killarney, Ireland, 12–17 July 2015; IEEE: New York, NY, USA, 2015; pp. 1–4.
40. Ma, K.; Duanmu, Z.; Wu, Q.; Wang, Z.; Yong, H.; Li, H.; Zhang, L. Waterloo exploration database: New challenges for image quality assessment models. *IEEE Trans. Image Process.* **2016**, *26*, 1004–1016.
41. Agustsson, E.; Timofte, R. Ntire 2017 challenge on single image super-resolution: Dataset and study. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 126–135.
42. Martin, D.; Fowlkes, C.; Tal, D.; Malik, J. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In Proceedings of the Eighth IEEE International Conference on Computer Vision (ICCV 2001), Vancouver, BC, Canada, 7–14 July 2001; IEEE: New York, NY, USA, 2001; Volume 2, pp. 416–423.
43. Roth, S.; Black, M.J. Fields of experts: A framework for learning image priors. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 21–25 June 2005; IEEE: New York, NY, USA, 2005; Volume 2, pp. 860–867.
44. Huang, J.B.; Singh, A.; Ahuja, N. Single image super-resolution from transformed self-exemplars. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 5197–5206.
45. Liang, J.; Cao, J.; Sun, G.; Zhang, K.; Van Gool, L.; Timofte, R. Swinir: Image restoration using swin transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 1833–1844.
46. Huynh-Thu, Q.; Ghanbari, M. Scope of validity of PSNR in image/video quality assessment. *Electron. Lett.* **2008**, *44*, 800–801.
47. Guo, X.; O'Neill, W.C.; Vey, B.; Yang, T.C.; Kim, T.J.; Ghassemi, M.; Pan, I.; Gichoya, J.W.; Trivedi, H.; Banerjee, I. SCU-Net: A deep learning method for segmentation and quantification of breast arterial calcifications on mammograms. *Med. Phys.* **2021**, *48*, 5851–5861.
48. Marmolin, H. Subjective MSE measures. *IEEE Trans. Syst. Man Cybern.* **1986**, *16*, 486–489.
49. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
50. Sahu, S.; Singh, A.K.; Ghrera, S.; Elhoseny, M. An approach for de-noising and contrast enhancement of retinal fundus image using CLAHE. *Opt. Laser Technol.* **2019**, *110*, 87–98.