DCTMamba: Advancing JPEG Image Restoration Through Long-Sequence Modeling and Adaptive Frequency Strategy

Xi Wang[1], Xueyang Fu[1]*, Liang Li[2], Zheng-Jun Zha[1]

[1]University of Science and Technology of China, Hefei, China
[2]Institute of Computing Technology, Chinese Academy of Sciences
wangxxi@mail.ustc.edu.cn; xyfu@ustc.edu.cn; liang.li@ict.ac.cn; zhazj@ustc.edu.cn

Abstract

Despite the advanced long-sequence modeling of Mamba, which has expanded its applications in image restoration, there remains a lack of exploration combining its strengths with the specific characteristics of JPEG image restoration, where high-frequency components are lost after the Discrete Cosine Transform (DCT). To address this, we introduce DCTMamba, a new framework designed to apply Mamba more effectively to JPEG image restoration. Specifically, our method integrates the Discrete Cosine Transform (DCT) into the Mamba to establish the sequential scanning from lower to higher frequencies, enabling the network to initially reconstruct coarse structures and progressively refine the image with more intricate details. Furthermore, recognizing the variable frequency distributions that arise from DCT transformations across different image sizes, we develop Scale-Adaptive Normalization to manage these variations adeptly. Comprehensive experiments confirm that DCTMamba outperforms existing solutions, achieving high fidelity in both coarse structures and fine details.

Code — https://github.com/wang-xi-1/Deblock25

Introduction

JPEG (Wallace 1992), a widely used lossy compression method for digital images, initially divides the image into 8x8 pixel blocks and then employs the Discrete Cosine Transform (DCT) (Khayam 2003) to shift these blocks from the spatial to the frequency domain. This transformation prioritizes the low-frequency components, significantly reducing high-frequency details that often encapsulate fine aspects of the image during quantization. This compression achieves substantial file size reduction, minimizing storage and bandwidth requirements. The Quality Factor (QF) in JPEG dictates how much high-frequency information is discarded, with lower QF values leading to stronger compression and more pronounced image distortion. While effective in reducing file size, JPEG compression introduces artifacts that degrade visual quality and hinder computer vision tasks like image classification(Peng et al. 2024) and object detection.

The swift advancements in deep learning have catalyzed a shift in the JPEG restoration domain, with neural network-based methods progressively superseding traditional model-based strategies (Foi, Katkovnik, and Egiazarian 2007; Zhang et al. 2012, 2013). These neural approaches have demonstrated exceptional efficacy, with convolutional neural networks (CNNs) like ARCNN (Dong et al. 2015), DnCNN (Zhang et al. 2017), and QGAC (Ehrlich et al. 2020a) employing their nonlinear mapping capabilities to transform degraded images back to their pristine states. Additionally, the approach proposed by Zhao *et al.* (Zhang et al. 2013) utilizes transformers to leverage its long-sequence modeling potential, thereby expanding the receptive fields significantly. More recently, Mamba has been at the forefront, particularly noted for its robust performance in long-sequence modeling (Gu, Goel, and Ré 2021; Gu and Dao 2023). Mamba is now utilized in image restoration projects to effectively utilize global information, setting a new standard in the field.

Nevertheless, these advanced methods encounter unique challenges: (1) CNN-based methods (Ehrlich et al. 2020a; Jiang, Zhang, and Timofte 2021; Fu et al. 2021) are limited by their inherent local reductive bias, resulting in suboptimal performance when addressing image restoration tasks that require global information; (2) Transformer-based methods (Liang et al. 2021) face quadratic computational complexity, making it challenging to handle very long sequence inputs, thus leading to bottlenecks in resource consumption and processing speed; (3) Although Mamba (Gu and Dao 2023) is suitable for causal autoregressive tasks (Yu and Wang 2024), it is unsuitable for non-causal image restoration tasks, restricting its effectiveness in such applications.

In this paper, we introduce DCTMamba, a new approach designed to address the challenges in JPEG restoration by synergizing JPEG compression traits with the capabilities of Mamba. Addressing the limitation of Mamba's sequence scanning, which is ill-suited for the non-causal dynamics of image data, we have integrated the Discrete Cosine Transform (DCT) into the system. This integration enables the scanning of sequences from low to high frequency, thereby establishing causal relationships within the sequences: beginning with a basic sketch and incrementally adding details, as shown in Fig. 1. Moreover, to effectively manage varying image sizes during the restoration process, we normal-

*Corresponding author

Figure 1: Direct pixel-by-pixel scanning in the spatial domain faces challenges due to the absence of causality. However, by applying the Discrete Cosine Transform (DCT) to the image and scanning from low to high frequencies, causality can be established. This means that the network can first outline the basic structure of the image and then gradually add texture details at the edges. The second and third columns show the low and high-frequency components after the DCT transformation.

ize the DCT coefficients. This normalization ensures a balanced contribution of different frequency components, leading to consistent restoration quality across different sizes and enhancing the detail and fidelity of the outputs. Recognizing the importance of local image features such as shapes, edges, and textures—which often have local correlations—we have also incorporated convolutional neural networks (CNNs) with Mamba. This integration allows the network to concurrently process both local and global information, optimizing the overall restoration effectiveness.

The contributions of this paper could be summarized as:

- We introduce the DCTMamba framework, which combines JPEG compression characteristics with Mamba's strengths. By integrating the Discrete Cosine Transform (DCT) into Mamba, it performs scans from low to high frequencies, establishing causal links between scan sequences and enhancing image restoration from coarse to fine details.
- To address the challenges posed by varying frequency component densities due to different image sizes, we introduce a normalization process for DCT coefficients. This normalization balances the contributions of various frequency components, ensuring consistent restoration quality across images of different sizes.
- We confirm the effectiveness of DCTMamba in restoring JPEG images through extensive tests on various datasets. Our results show that DCTMamba exceeds current state-of-the-art methods, highlighting its superior restoration abilities.

Related Works

JPEG Image Restoration

ARCNN, proposed by Dong *et al.* (Dong et al. 2015), is the first work to use CNNs for JPEG image restoration, achieving good performance with just four layers. Wang *et al.* (Wang et al. 2016) introduce DCT domain priors based on JPEG compression characteristics, further enhancing the restoration effects. Mao *et al.* (Mao, Shen, and Yang 2017) deepen the network design and propose a deep encoding-decoding network that better utilizes deep features. Subsequently, traditional methods like multi-scale constraints (Cavigelli, Hager, and Benini 2017) and wavelet signal structures (Liu et al. 2018) are also integrated into deep neural networks. With the advent of residual structures, the depth of neural networks further increases. Zhang *et al.* (Zhang et al. 2017) accelerate network convergence and achieve blind JPEG image restoration by incorporating Batch Normalization (BN) (Ioffe and Szegedy 2015) and residual learning (He et al. 2016). Ehrlich *et al.* (Ehrlich et al. 2020b) utilize GAN loss to assist the JPEG image restoration network, generating images with more realistic details, taking advantage of the generative model's superior capability in texture detail generation. To more effectively handle JPEG images of varying compression qualities with a single network, several works consider incorporating the compression factor as auxiliary information. Kim *et al.* (Kim et al. 2019) estimate the compression factor before performing JPEG image restoration. AGARNet (Kim, Soh, and Cho 2020) achieves finer granularity by estimating pixel-wise quality factors, thus covering a wide range of quality factors with a single network. Jiang *et al.* (Jiang, Zhang, and Timofte 2021) use a supervised approach to predict the compression quality factor directly and embed the predicted factor into subsequent networks to guide JPEG artifact removal. In order to obtain a larger receptive field, Zhao *et al.* (Zhao et al. 2023) propose an efficient image restoration Transformer that captures global dependencies at the superpixel level before transferring them to individual pixels. However, some of these methods fail to model the global receptive fields, while others have quadratic complexity, leading to the limitations of these approaches.

State Space Models

State Space Models (SSMs) (Gu, Goel, and Ré 2021; Gu et al. 2022; Gu and Dao 2023) garner significant attention due to their ability to linearly scale with sequence length in long-range dependency modeling. The Structured State-Space Sequence model (S4) (Gu, Goel, and Ré 2021) is among the first to employ deep state-space models for this purpose. Building on this, the S5 layer (Smith, Warrington, and Linderman 2022) introduces MIMO SSM and efficient parallel scanning. Mehta *et al.* (Mehta et al. 2022) further enhance S4 by incorporating the Gated State Space layer. To address the issue that the current system remains static for varying inputs, Mamba (Gu and Dao 2023) introduces data dependency into State Space Models, allowing the model parameters to respond differently to different inputs. VMamba (Liu et al. 2024) addresses the gap between sequentially ordered data and non-causal visual images. Vim (Zhu et al. 2024) introduces a bidirectional state space model with positional awareness. LocalMamba (Huang et al. 2024) focuses on a local scanning method to preserve local context dependencies. EfficientVMamba (Pei, Huang, and Xu 2024) is designed as a lightweight SSM, incorporating an additional convolution branch to learn both global and lo-

cal representational features. MambaIR (Guo et al. 2024) enhances Mamba's capabilities by utilizing convolution and channel attention mechanisms. However, these methods do not address the issue of sequentially ordered data and non-causal visual images. We introduce the DCT into Mamba, making it more suitable for JPEG image restoration tasks.

Methodology

The recently proposed Mamba attracts widespread interest due to its linear complexity. This work introduces the discrete cosine transform (DCT) into the Mamba. First, we review the theoretical background of the image DCT and the Mamba. Then, we describe in detail how to combine them to better serve the JPEG image restoration task.

Background

Image Discrete Cosine Transform. The Discrete Cosine Transform (DCT) (Khayam 2003) converts an image from the spatial domain to the frequency domain as a sum of cosine functions oscillating at different frequencies. The forward DCT for an image of size $N \times N$ is defined as:

$$F(u,v) = \frac{2}{N} \sum_{x=0}^{N-1} \cos\left(\frac{(2x+1)u\pi}{2N}\right) \left(\sum_{y=0}^{N-1} f(x,y) \cos\left(\frac{(2y+1)v\pi}{2N}\right)\right), \tag{1}$$

the inverse DCT, which reconstructs the image from its frequency coefficients $F(u,v)$, is defined as:

$$f(x,y) = \frac{2}{N} \sum_{u=0}^{N-1} \cos\left(\frac{(2x+1)u\pi}{2N}\right) \left(\sum_{v=0}^{N-1} F(u,v) \cos\left(\frac{(2y+1)v\pi}{2N}\right)\right), \tag{2}$$

where $f(x,y)$ is the pixel value at position (x,y) in the spatial domain, $F(u,v)$ is the DCT coefficient at frequency coordinates (u,v) in the frequency domain.

The low-frequency components at the top-left (u, v coordinates are small) contain the most significant information about the image, while the high-frequency components at the bottom-right (u, v coordinates are large) primarily represent the texture details. The process of JPEG compression involves discarding high frequencies through quantization. Therefore, the task of restoring JPEG images is to recover these compressed high-frequency components.

Meanwhile, the frequency basis vectors of the DCT are influenced by the image size. Consequently, images of different sizes exhibit different frequency coefficient distribution densities after the DCT. Therefore, when using DCT for image restoration, it is essential to consider the image size to appropriately handle the varying levels of detail and texture information represented in the DCT coefficients.

State Space Models. The structured state-space sequence models (S4) (Gu, Goel, and Ré 2021) are predominantly influenced by recent advancements in continuous linear time-invariant (LTI) (Willems 1986) systems. These systems map a one-dimensional sequence or function $x(t) \in \mathbb{R}$ to $y(t) \in \mathbb{R}$ via a hidden state $h(t) \in \mathbb{R}^N$. This can formally be described by the following linear ordinary differential equation (ODE):

$$\begin{aligned} h'(t) &= Ah(t) + Bx(t), \\ y(t) &= Ch(t) + Dx(t), \end{aligned} \tag{3}$$

where N denotes the state size, and the matrices $\mathbf{A} \in \mathbb{R}^{N\times N}$, $\mathbf{B} \in \mathbb{R}^{N\times 1}$, $\mathbf{C} \in \mathbb{R}^{1\times N}$, and the scalar $\mathbf{D} \in \mathbb{R}$.

To make Eq. (3) applicable in deep learning, a discretization step is typically performed, and the timescale parameter Δ is used to convert the continuous parameters $\mathbf{A}, \mathbf{B}$ into their discrete counterparts. One common discretization technique is the zero-order hold (ZOH) rule, which can be expressed as:

$$\begin{aligned} \bar{A} &= \exp(\Delta A), \\ \bar{B} &= (\Delta A)^{-1}(\exp(\Delta A) - I) \cdot \Delta B. \end{aligned} \tag{4}$$

After discretization, Eq. (3) is converted into its discrete form for a step size Δ as follows:

$$\begin{aligned} h_k &= \bar{A}h_{k-1} + \bar{B}x_k, \\ y_k &= Ch_k + Dx_k. \end{aligned} \tag{5}$$

However, the formulation in Eq. (5) addresses an LTI system with fixed parameters for varying inputs. To overcome this limitation, Mamba (Gu and Dao 2023) suggests integrating a selective scan mechanism, where the matrices $\bar{\mathbf{B}}_k$, $\bar{\mathbf{C}}_k$, and Δ_k are computed from the input data. The input-adaptive parameter adjustment can be described as:

$$\begin{aligned} \bar{\mathbf{B}}_\mathbf{k} &= f_B(x_k), \\ \bar{\mathbf{C}}_\mathbf{k} &= f_C(x_k), \\ \Delta_k &= \theta_A(P + f_A(x_k)), \end{aligned} \tag{6}$$

where $f_B(x_k)$, $f_C(x_k)$, and $f_A(x_k)$ are linear functions that expand the features to the hidden state dimensions.

Due to the non-causal nature of JPEG images, the temporal characteristics of the scanning mechanism are not well-suited for JPEG image restoration tasks. To address this issue, we introduce image DCT in Mamba to establish causal relationships in the image.

Overall Pipeline

We first introduce the overall framework of our DCT-Mamba, as shown in Fig. 2. Our network's architecture is based on the U-net model, where the foundational building block is a Global-to-Local Block (GLB) composed of Mamba modules for global information reconstruction and convolutional layers for local information reconstruction. Due to the nature of JPEG image compression, which discards high-frequency information after the DCT transformation and retains low-frequency information, and the fact that image data lacks causality, we propose a Coarse-to-Fine State Space Block (CFSSB). By incorporating the DCT into

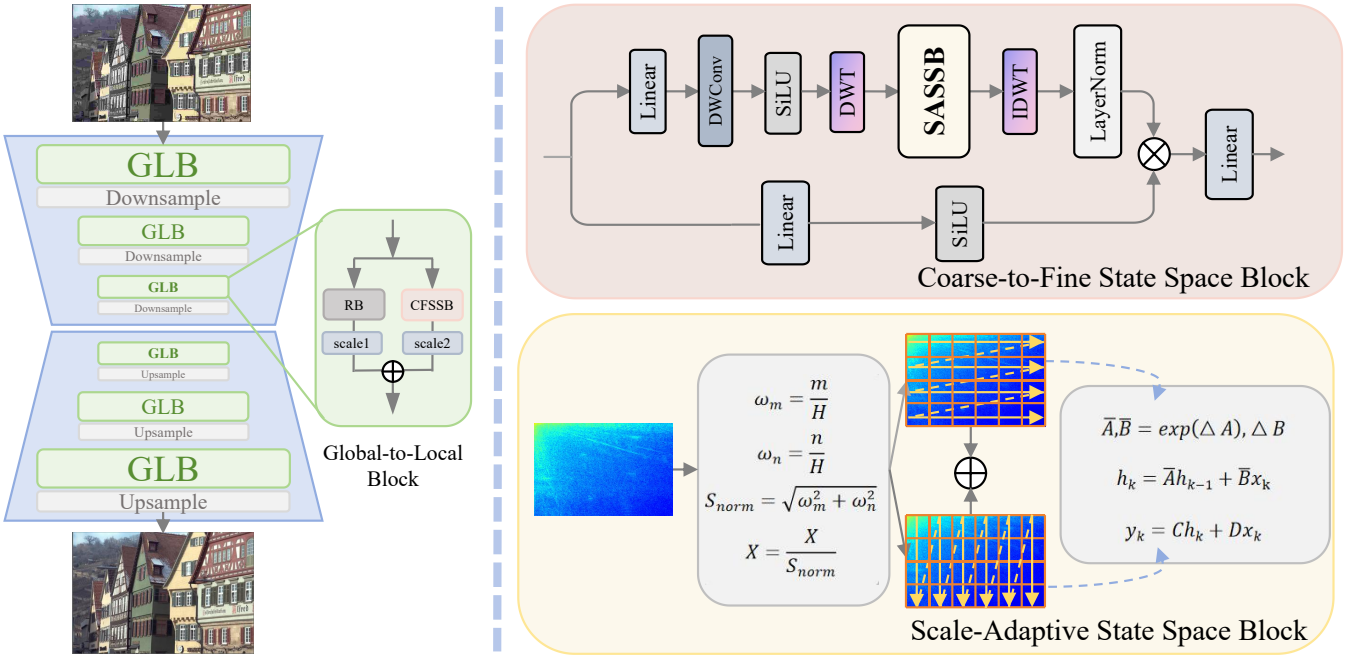


Figure 2: The overall architecture of our proposed DCTMamba for image restoration includes the following key components: Global-to-Local Block, Coarse-to-Fine State Space Block and Scale-Adaptive State Space Block.

the Mamba, we set the scan order from low to high frequencies, which is inherently causal. This allows the network first to generate the coarse structure and outlines of the image, followed by the edges and finer details. And since the frequency coefficient distribution densities of the DCT image transformation are determined solely by the image sizes, we propose the Scale-Adaptive State Space Block (SASSB) to make the network more suitable for inputs of any size. We describe the design and utility of these modules in detail in the following sections.

Global-to-Local Block. As illustrated in Fig. 2, the Global-to-Local Block module consists of two main components: the Coarse-to-Fine State Space Module (CFSSM) for global information reconstruction, and ResBlock (RB) for local information reconstruction through convolution layers. These two branches operate in parallel, each independently processing the input data. The outputs from both branches are then combined by adding them together, with learned weights applied to each output. This process can be mathematically expressed by the following formula:

$$y = s_1 CFSSM(X) + s_2 RB(X), \tag{7}$$

where $X \in \mathbb{R}^{H\times W\times C}$ are the input features, $y \in \mathbb{R}^{H\times W\times C}$ is the output, and $s_1, s_2 \in \mathbb{R}^{1\times 1\times C}$ are the learned weights.

Coarse-to-Fine State Space Block. In this module, we integrate Mamba's sequence scanning mechanism with the characteristics of JPEG image compression. First, we convert the input features using the Discrete Cosine Transform (DCT) to decompose the image into different frequency components. Next, we scan these components in sequence, starting from the low frequencies and moving to the high frequencies. By focusing on low frequencies first, we outline the basic shapes and structures of the image. As we progress to higher frequencies, we add details and textures. This coarse-to-fine approach helps us construct the image from the main features to the finer details.

Scale-Adaptive State Space Block. Since the frequency components distribution densities of the DCT transform depending on the image size, different input sizes will yield different DCT results, affecting the transform's consistency. To address this issue, we employ a method to normalize the DCT coefficients after the DCT transform, thereby eliminating the impact introduced by size differences. Specifically, we normalize the DCT coefficients according to the image size. The formulas are as follows:

$$\begin{aligned}
\omega_m &= \frac{m}{H}, \quad m = \{0, 1, \ldots, H-1\}^T, \\
\omega_n &= \frac{n}{W}, \quad n = \{0, 1, \ldots, W-1\}, \\
S_{\text{norm}} &= \begin{cases} \sqrt{\omega_m^2 + \omega_n^2}, & \text{if } (m,n) \neq (0,0), \\ 1, & \text{if } (m,n) = (0,0), \end{cases} \\
X_{norm} &= \frac{X}{S_{\text{norm}}},
\end{aligned} \tag{8}$$

where m and n represent the indices of the DCT coefficients, corresponding to the height (H) and width (W) of the image, respectively. The terms ω_m and ω_n are the normalized indices, scaled according to the image dimensions. The normalization factor (norm) is calculated to proportionally adjust each DCT coefficient. This operation normalizes the frequency components to eliminate the impact of image size

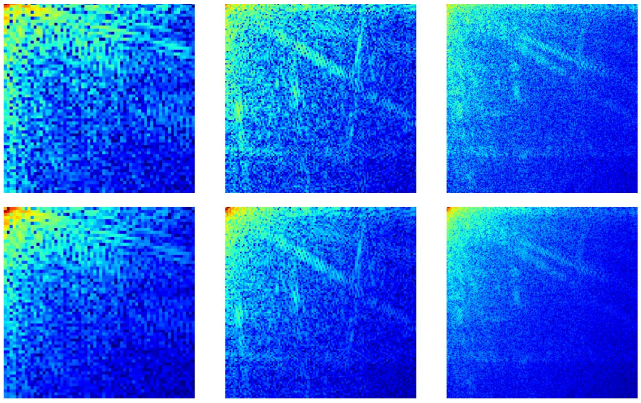


Figure 3: To demonstrate the impact of different sizes on image DCT, we crop patches of sizes 64, 128, and 256 (from left to right) from the same image. The top row shows the DCT results before normalization, while the bottom row displays the results after normalization. After normalization, the distribution of DCT coefficients becomes more uniform.

differences on DCT results, ensuring that images of different sizes achieve consistent results after the DCT transformation. This ensures that subsequent processing steps have the same effectiveness for images of different sizes.

Discussion

To more clearly articulate the rationale behind our network design, we will delve into an in-depth discussion of the three proposed module design schemes.

In images, the local relationships between pixels are vital. While Mamba (Gu and Dao 2023) handles long-range dependencies, it lacks a mechanism to emphasize local information, potentially missing detailed features. Convolutional Neural Networks (CNNs) excel at modeling local patterns, thanks to their translational invariance, allowing them to capture local features. Integrating CNNs into Mamba can address these shortcomings, allowing the model to leverage both long-range dependencies and local features, thereby enhancing overall performance.

Images do not have explicit causal relationships, so using Mamba's original sequential scanning mechanism for image restoration is unreasonable. In restoring images, we first generate the main contours and structures (low-frequency components) and then gradually refine them to add texture details (high-frequency components). Additionally, since JPEG compression discards high-frequency components following DCT, we apply the DCT to the image and adopt a scanning method from low-frequency to high-frequency, achieving a coarse-to-fine image restoration process. This method not only better models the causal relationships between sequence orders but also allows for gradual refinement and enhancement of detail quality.

As shown in Eq. (1), the size of an image directly affects the density and distribution of frequency components after the DCT. The summation terms $\sum_{x=0}^{N-1}$ and $\sum_{y=0}^{N-1}$ represent the traversal of image pixels, meaning the number of pixels influences the frequency component density. The frequencies of the cosine functions, $\cos\left(\frac{(2x+1)u\pi}{2N}\right)$ and $\cos\left(\frac{(2y+1)v\pi}{2N}\right)$, depend on $\frac{u\pi}{2N}$ and $\frac{v\pi}{2N}$, respectively. As N increases, the frequencies $\frac{u\pi}{2N}$ and $\frac{v\pi}{2N}$ become smaller, allowing for the representation of finer frequency details. To handle images with varying resolutions during the restoration process, normalizing the DCT coefficients is necessary. We conduct qualitative visual analysis, with results shown in Fig. 3. The figure displays the distribution of DCT coefficients before and after applying the Scale-Adaptive norm operation, demonstrating that this operation leads to a more balanced distribution of DCT coefficients.

Loss Function

We employed two distinct loss functions to optimize our network: the Charbonnier loss function (Lai et al. 2018), the DCT loss function and the SSIM (Wang et al. 2004) loss function. where Y_{pred} and Y_{gt} represent the predicted and ground truth values, respectively, and N is the number of images in a mini-batch.

The Charbonnier loss function is defined as:

$$L_{char} = \frac{1}{N}\sum_{i=1}^{N}\sqrt{(Y_{pred,i} - Y_{gt,i})^2 + \epsilon}, \tag{9}$$

where ϵ is a small constant (set to 1e-3) added for numerical stability.

The DCT loss function is formulated as:

$$L_{dct} = \frac{1}{N}\sum_{i=1}^{N}\left|\mathrm{DCT}(Y_{pred,i}) - \mathrm{DCT}(Y_{gt,i})\right|, \tag{10}$$

where $\mathrm{DCT}(Y_{pred})$ and $\mathrm{DCT}(Y_{gt})$ denote the Discrete Cosine Transform of the predicted and ground truth values, respectively.

The SSIM loss function is formulated as:

$$L_{ssim} = 1 - \frac{1}{N}\sum_{i=1}^{N}\mathrm{SSIM}(Y_{pred,i}, Y_{gt,i}). \tag{11}$$

The total loss is then given by:

$$L_{total} = \lambda_{char}L_{char} + \lambda_{dct}L_{dct} + \lambda_{ssim}L_{ssim}, \tag{12}$$

where the coefficients are set to $\lambda_{char} = 1$, $\lambda_{dct} = 0.1$, and $\lambda_{ssim} = 0.2$, respectively.

Experiments

Datasets. In our experiments, for the training phase, we use a combination of the following datasets: DIV2K (800 training images) (Agustsson and Timofte 2017), Flickr2K (2650 images) (Timofte et al. 2017), BSD500 (200 training images) (Arbelaez et al. 2011), and WED (4744 images). For testing, we use Classic5 (Zeyde, Elad, and Protter 2010), LIVE1 (Sheikh 2005) and the test sets of BSDS00 (Arbelaez et al. 2011). In both the training and testing phases, we utilize the Y channel of the YCbCr space, mixing data from Q10 to Q80 in increments of 10 during training, and using Q10, Q20, Q30, and Q40 for testing.

Dataset	Quality	JPEG	ARCNN	DnCNN	DCSC	MWCNN
$Classic5$	10	27.82/0.769/27.78	29.03/0.793/28.76	29.40/0.803/29.13	29.62/0.810/29.30	30.01/0.820/29.59
	20	30.12/0.845/30.05	31.15/0.852/30.59	31.63/0.861/31.19	31.81/0.864/31.34	32.16/0.870/31.52
	30	31.48/0.876/31.37	32.51/0.881/31.98	32.91/0.886/32.38	33.06/0.888/32.49	33.43/0.893/32.62
	40	32.43/0.894/32.28	33.32/0.895/32.79	33.77/0.900/33.23	33.87/0.902/33.30	34.27/0.906/33.35
$LIVE1$	10	27.77/0.780/27.72	28.96/0.808/28.68	29.19/0.812/28.90	29.34/0.818/29.01	29.69/0.825/29.32
	20	30.07/0.860/29.99	31.29/0.873/30.76	31.59/0.880/31.07	31.70/0.883/31.18	32.04/0.889/31.51
	30	31.41/0.893/31.30	32.67/0.904/32.14	32.98/0.909/32.34	33.07/0.911/32.43	33.45/0.915/32.80
	40	32.36/0.911/32.22	33.61/0.920/33.11	33.96/0.925/33.28	34.02/0.926/33.36	34.45/0.930/33.78
$BSDS500$	10	27.80/0.768/25.10	29.10/0.804/28.73	29.21/0.809/28.80	29.32/0.813/28.91	29.61/0.820/29.14
	20	30.05/0.849/27.22	31.28/0.870/30.55	31.53/0.878/30.79	31.63/0.880/30.92	31.92/0.885/31.15
	30	31.37/0.884/28.53	32.67/0.902/31.94	32.90/0.907/31.97	32.99/0.908/32.08	33.30/0.912/32.34
	40	32.30/0.903/29.49	33.55/0.918/32.78	33.85/0.923/32.80	33.92/0.924/32.92	34.27/0.928/33.19

Dataset	Quality	RDN	QGAC	FBCNN	Zhao *et al.*	**Ours**
$Classic5$	10	30.03/0.819/29.59	29.84/0.812/25.21	30.12/0.822/29.80	30.12/0.822/29.80	**30.25/0.824/29.91**
	20	32.19/0.870/31.53	31.98/0.869/27.50	32.31/0.872/31.74	32.35/0.873/31.76	**32.44/0.874/31.85**
	30	33.46/0.893/32.59	33.22/0.892/28.94	33.54/0.894/32.78	33.59/0.895/32.80	**33.64/0.895/32.84**
	40	-	34.05/0.905/29.92	34.35/0.907/33.48	34.41/0.907/33.49	**34.46/0.908/33.53**
$LIVE1$	10	29.70/0.825/29.37	29.51/0.825/25.33	29.75/0.827/29.40	29.78/0.827/29.45	**29.85/0.829/29.50**
	20	32.10/0.889/31.29	31.83/0.888/27.57	32.13/0.889/31.57	32.15/0.889/31.60	**32.23/0.891/31.69**
	30	33.54/0.916/32.62	33.20/0.914/28.92	33.54/0.916/32.83	33.57/0.916/32.87	**33.65/0.917/32.95**
	40	-	34.16/0.929/29.96	34.53/0.931/33.74	34.55/0.931/33.78	**34.63/0.932/33.87**
$BSDS500$	10	29.24/0.808/28.71	29.46/0.821/25.10	29.67/0.821/29.22	29.65/0.821/29.15	**29.76/0.824/29.31**
	20	31.48/0.879/30.45	31.73/0.884/27.22	32.00/0.885/31.19	31.98/0.885/31.11	**32.09/0.887/31.29**
	30	32.83/0.908/31.60	33.07/0.912/28.53	33.37/0.913/32.32	33.37/0.913/32.25	**33.47/0.914/32.43**
	40	-	34.01/0.927/29.49	34.33/0.928/33.10	34.34/0.928/33.05	**34.43/0.929/33.22**

Table 1: Quantitative comparisons of different methods on **grayscale** JPEG images. PSNR↑ / SSIM↑ / PSNR-B↑ format. The best results are **boldfaced**.

Implementation Details. Our experiments are implemented in PyTorch and executed on four NVIDIA GTX 4090 GPUs. For data preparation, images are initially randomly shuffled and resized into training patches of size 256×256. Data augmentation methods such as rotation and flipping are employed to enhance variability. We optimize the network using the Adam algorithm (Kingma and Ba 2014) with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate starts at 2×10^{-4} and is gradually decreased to 1×10^{-6} using a cosine annealing schedule. We select PSNR, SSIM (Wang et al. 2004), and PSNR-B (Tadala and Narayana 2012) as the metrics for evaluating performance.

Quantitative Comparison. As illustrated in Tab. 1, we assess our approach's performance against state-of-the-art methods for JPEG image restoration. Our method employs a single weight that effectively covers JPEG compression levels from 10 to 40, demonstrating superior results in PSNR, SSIM, and PSNRB across three benchmark datasets. This showcases its effectiveness and robustness compared to existing techniques.

Qualitative Comparison. In addition, we showcase the visual impact of our method in Fig. 4 and Fig. 5, including synthetic and real-world datasets. In the real-world dataset, we employ residuals for visualization. The visualization results indicate that our method is more successful at recovering texture details in JPEG-compressed images, thereby underscoring its effectiveness.

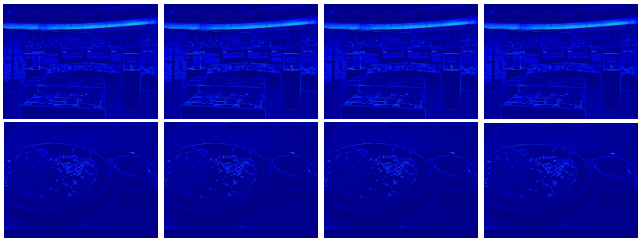
Ablation Studies

We conduct ablation experiments on the Q40 of the Classic5 dataset to verify the effectiveness of our method.

Global-to-Local Block. In order to verify the effectiveness of the local module within the Global-to-Local Block, we remove the Mamba module from this block, referring to it as 'Baseline'. Additionally, we remove the convolution module, which we refer to as 'Model-1'. As shown in Tab. 2, quantitative analysis indicates that the network containing both local and global modules performs better.

JPEG ARCNN DnCNN RNAN FBCNN Zhao *et al.* Ours GT

Figure 4: Visual comparisons on the 'Classic5' dataset.



Twitter ARCNN FBCNN Ours

Figure 5: We perform visual comparisons on real-world images from the 'Twitter' dataset, employing residual images for visualization. It can be seen that our method yields smaller residuals in edge details.

Method	LB	GB	CFSSB	SASSB	PSNR↑ / SSIM↑
Baseline	✓				34.21 / 0.905
Model-1		✓	✓	✓	34.33 / 0.906
Model-2	✓	✓			34.29 / 0.906
Model-3	✓	✓	✓		34.37 / 0.907
Ours	✓	✓	✓	✓	34.46 / 0.908

Table 2: Ablation studies on the design of blocks.

Coarse-to-Fine State Space Block. To verify the superiority of our proposed Coarse-to-Fine State Space Block for scanning in the DCT frequency domain over direct time-domain scanning, we conduct an experiment using direct time-domain scanning, referred to as 'Model-2'. The results demonstrate that scanning in the DCT domain offers enhanced performance.

Scale-Adaptive State Space Block. To validate the effectiveness of our proposed Scale-Adaptive State Space Block, we remove this operation from the network and refer to it as 'Model-3'. Our quantitative experimental analysis shows that the network's performance somewhat declines after removing this operation.

Investigation of the Loss Function. We further explore how various combinations of loss functions impact performance. Specifically, we set λ_{dct} and λ_{ssim} to 0, respectively, to assess the contribution of each loss function. The quantitative outcomes, presented in Tab. 3, demonstrate that both loss functions play a role in enhancing performance.

No.	λ_{char}	λ_{dct}	λ_{ssim}	PSNR↑ / SSIM↑
1	1	1	0	34.42 / 0.906
2	1	0	1	34.39 / 0.907
3	1	1	1	34.46 / 0.908

Table 3: Quantitative results of different loss combinations. Here, '1' indicates the use of this loss function.

Method	ARCNN	DnCNN	RDN	FBCNN	Zhao et al.	Ours
Param (M)	0.11	0.56	22.12	71.9	12.18	16.60
PSNR (dB)	29.03	29.40	30.03	30.12	30.12	30.25

Table 4: Comparison of parameters and PSNR in Classic5 Q10 across different methods.

Computational Comparison. We compare our method with previous approaches in Tab. 4 regarding parameters and PSNR values. The results clearly show that our method strikes a good balance between performance and effectiveness, achieving competitive PSNR scores while maintaining a reasonable number of parameters when compared to other methods in the field.

Conclusion

In this paper, we present DCTMamba, a novel framework that effectively combines the strengths of Mamba with the specific characteristics of JPEG compression. By integrating the Discrete Cosine Transform (DCT) into the network, DCTMamba addresses the challenge of non-causal image reconstruction, which Mamba alone could not resolve. Our approach establishes a causal relationship in the reconstruction process by scanning sequences from low to high frequencies. This integration allows the framework first to reconstruct coarse structures and progressively refine fine details. Additionally, to manage the diverse frequency distributions resulting from DCT transformation at different image sizes, we introduce Scale-Adaptive Normalization, enabling efficient handling of varying image sizes. Extensive experiments validate the effectiveness of our proposed approach.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) under Grants 62225207, 62436008, 62422609 and 62276243.

References

Agustsson, E.; and Timofte, R. 2017. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 126–135.

Arbelaez, P.; Maire, M.; Fowlkes, C.; and Malik, J. 2011. Contour Detection and Hierarchical Image Segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(5): p.898–916.

Cavigelli, L.; Hager, P.; and Benini, L. 2017. CAS-CNN: A deep convolutional neural network for image compression artifact suppression. In *2017 International Joint Conference on Neural Networks (IJCNN)*, 752–759. IEEE.

Dong, C.; Deng, Y.; Loy, C. C.; and Tang, X. 2015. Compression artifacts reduction by a deep convolutional network. In *Proceedings of the IEEE international conference on computer vision*, 576–584.

Ehrlich, M.; Davis, L.; Lim, S.-N.; and Shrivastava, A. 2020a. Quantization guided jpeg artifact correction. In *European Conference on Computer Vision*, 293–309. Springer.

Ehrlich, M.; Davis, L.; Lim, S.-N.; and Shrivastava, A. 2020b. Quantization guided jpeg artifact correction. In *European Conference on Computer Vision*, 293–309. Springer.

Foi, A.; Katkovnik, V.; and Egiazarian, K. 2007. Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images. *IEEE transactions on image processing*, 16(5): 1395–1411.

Fu, X.; Wang, M.; Cao, X.; Ding, X.; and Zha, Z.-J. 2021. A model-driven deep unfolding method for jpeg artifacts removal. *IEEE Transactions on Neural Networks and Learning Systems*, 33(11): 6802–6816.

Gu, A.; and Dao, T. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

Gu, A.; Goel, K.; Gupta, A.; and Ré, C. 2022. On the parameterization and initialization of diagonal state space models. *Advances in Neural Information Processing Systems*, 35: 35971–35983.

Gu, A.; Goel, K.; and Ré, C. 2021. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*.

Guo, H.; Li, J.; Dai, T.; Ouyang, Z.; Ren, X.; and Xia, S.-T. 2024. Mambair: A simple baseline for image restoration with state-space model. *arXiv preprint arXiv:2402.15648*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Huang, T.; Pei, X.; You, S.; Wang, F.; Qian, C.; and Xu, C. 2024. Localmamba: Visual state space model with windowed selective scan. *arXiv preprint arXiv:2403.09338*.

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, 448–456. PMLR.

Jiang, J.; Zhang, K.; and Timofte, R. 2021. Towards flexible blind JPEG artifacts removal. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4997–5006.

Khayam, S. A. 2003. The discrete cosine transform (DCT): theory and application. *Michigan State University*, 114(1): 31.

Kim, Y.; Soh, J. W.; and Cho, N. I. 2020. AGARNet: adaptively gated JPEG compression artifacts removal network for a wide range quality factor. *IEEE Access*, 8: 20160–20170.

Kim, Y.; Soh, J. W.; Park, J.; Ahn, B.; Lee, H.-S.; Moon, Y.-S.; and Cho, N. I. 2019. A pseudo-blind convolutional neural network for the reduction of compression artifacts. *IEEE Transactions on circuits and systems for video technology*, 30(4): 1121–1135.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lai, W.-S.; Huang, J.-B.; Ahuja, N.; and Yang, M.-H. 2018. Fast and accurate image super-resolution with deep laplacian pyramid networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(11): 2599–2613.

Liang, J.; Cao, J.; Sun, G.; Zhang, K.; Van Gool, L.; and Timofte, R. 2021. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, 1833–1844.

Liu, P.; Zhang, H.; Zhang, K.; Lin, L.; and Zuo, W. 2018. Multi-level wavelet-CNN for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 773–782.

Liu, Y.; Tian, Y.; Zhao, Y.; Yu, H.; Xie, L.; Wang, Y.; Ye, Q.; and Liu, Y. 2024. VMamba: Visual State Space Model. *ArXiv*, abs/2401.10166.

Mao, X. J.; Shen, C.; and Yang, Y. B. 2017. Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections. *IEEE transactions on image processing.*, 15(13): 3142–3155.

Mehta, H.; Gupta, A.; Cutkosky, A.; and Neyshabur, B. 2022. Long range language modeling via gated state spaces. *arXiv preprint arXiv:2206.13947*.

Pei, X.; Huang, T.; and Xu, C. 2024. Efficientvmamba: Atrous selective scan for light weight visual mamba. *arXiv preprint arXiv:2403.09977*.

Peng, L.; Cao, Y.; Sun, Y.; and Wang, Y. 2024. Lightweight Adaptive Feature De-drifting for Compressed Image Classification. *IEEE Transactions on Multimedia*.

Sheikh, H. 2005. LIVE image quality assessment database release 2. *http://live.ece.utexas.edu/research/quality*.

Smith, J. T.; Warrington, A.; and Linderman, S. W. 2022. Simplified state space layers for sequence modeling. *arXiv preprint arXiv:2208.04933*.

Tadala, T.; and Narayana, S. E. V. 2012. A Novel PSNR-B Approach for Evaluating the Quality of De-blocked Images.

Timofte, R.; Agustsson, E.; Van Gool, L.; Yang, M.-H.; and Zhang, L. 2017. Ntire 2017 challenge on single image super-resolution: Methods and results. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 114–125.

Wallace, G. K. 1992. The JPEG still picture compression standard. *IEEE transactions on consumer electronics*, 38(1): xviii–xxxiv.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4): 600–612.

Wang, Z.; Liu, D.; Chang, S.; Ling, Q.; Yang, Y.; and Huang, T. S. 2016. D3: Deep dual-domain based fast restoration of JPEG-compressed images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2764–2772.

Willems, J. C. 1986. From time series to linear system—Part I. Finite dimensional linear time invariant systems. *Automatica*, 22(5): 561–580.

Yu, W.; and Wang, X. 2024. MambaOut: Do We Really Need Mamba for Vision? *arXiv preprint arXiv:2405.07992*.

Zeyde, R.; Elad, M.; and Protter, M. 2010. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, 711–730. Springer.

Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; and Zhang, L. 2017. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 26(7): 3142–3155.

Zhang, X.; Xiong, R.; Fan, X.; Ma, S.; and Gao, W. 2013. Compression artifact reduction by overlapped-block transform coefficient estimation with block similarity. *IEEE transactions on image processing*, 22(12): 4613–4626.

Zhang, X.; Xiong, R.; Ma, S.; and Gao, W. 2012. Reducing blocking artifacts in compressed images via transform-domain non-local coefficients estimation. In *2012 IEEE International Conference on Multimedia and Expo*, 836–841. IEEE.

Zhao, H.; Gou, Y.; Li, B.; Peng, D.; Lv, J.; and Peng, X. 2023. Comprehensive and delicate: An efficient transformer for image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 14122–14132.

Zhu, L.; Liao, B.; Zhang, Q.; Wang, X.; Liu, W.; and Wang, X. 2024. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*.