mathematics

Article

Memory-Efficient Discrete Cosine Transform Domain Weight Modulation Transformer for Arbitrary-Scale Super-Resolution

Min Hyuk Kim and Seok Bong Yoo *

Deparment of Artificial Intelligence Convergence, Chonnam National University,
Gwangju 61186, Republic of Korea; min_hyuk@jnu.ac.kr
* Correspondence: sbyoo@jnu.ac.kr; Tel.: +82-625303437

Abstract: Recently, several arbitrary-scale models have been proposed for single-image super-resolution. Furthermore, the importance of arbitrary-scale single image super-resolution is emphasized for applications such as satellite image processing, HR display, and video-based surveillance. However, the baseline integer-scale model must be retrained to fit the existing network, and the learning speed is slow. This paper proposes a network to solve these problems, processing super-resolution by restoring the high-frequency information lost in the remaining arbitrary-scale while maintaining the baseline integer scale. The proposed network extends an integer-scaled image to an arbitrary-scale target in the discrete cosine transform spectral domain. We also modulate the high-frequency restoration weights of the depthwise multi-head attention to use memory efficiently. Finally, we demonstrate the performance through experiments with existing state-of-the-art models and their flexibility through integration with existing integer-scale models in terms of peak signal-to-noise ratio (PSNR) and similarity index measure (SSIM) scores. This means that the proposed network restores high-resolution (HR) images appropriately by improving the image sharpness of low-resolution (LR) images.

Keywords: machine learning; pattern recognition; arbitrary scale super-resolution; DCT spectral domain; multi-head self attention; vision transformer

MSC: 68T45

check for updates

Citation: Kim, M.H.; Yoo, S.B. Memory-Efficient Discrete Cosine Transform Domain Weight Modulation Transformer for Arbitrary-Scale Super-Resolution. *Mathematics* **2023**, *11*, 3954. https://doi.org/10.3390/math11183954

Academic Editor: Valsamis Ntouskos

Received: 17 August 2023
Revised: 15 September 2023
Accepted: 16 September 2023
Published: 18 September 2023

1. Introduction

We purpose arbitrary-scale super-resolution (SR), which upsamples decimal (floating point) scale in single-image SR (SISR). SISR, which aims to recover high-resolution (HR) images from low-resolution (LR) images, is a recent research study in traditional computer vision and has immense potential in various applications, such as video games, satellite images, medical images, surveillance, monitoring, video enhancement, and security. In addition, SISR has also been recognized as a challenging task due to its ill-posed nature, and various methods have been proposed [1–4]. Moreover, SISR methods, called integer-scale super-resolution (SR) [1–4], train on the characteristics of the LR image and upsample the LR image with a fixed upsampling layer. Most integer-scale SISR methods consist of a deep neural network (DNN) and an upsampling layer called pixel shuffling. The limitation of these pixel shuffling modules is that they cannot generate SR images at a noninteger scale. Therefore, many proposed approaches require a separate DNN model for each upsampling layer, usually restricted to a limited number of integers (e.g., ×2, ×3, or ×4). DNN models trained with fixed integer-scale upsampling layers are challenging to perform arbitrary-scale SR because they only perform fixed integer-scale SR. Furthermore, these separate DNN models trained in this way ignore the SR correlation at different scales, leading to discontinuous representation and limited performance. In addition, the memory problem of storing each integer-scale factor and a wide range of arbitrary-scales in real-world scenarios makes selecting and applying SISR models problematic in practical applications. These shortcomings limit their applicability and flexibility in real-world scenarios. An approach to arbitrary-scale SR has emerged and received considerable attention to address the limitations. We note that the arbitrary-scale denotes floating point scale.

For example, in aspects of display applications, utilizing an arbitrary-scale SR can fit an HR image for any size input image. Moreover, it is a common requirement to arbitrarily zoom in on an image by rolling a mouse wheel. Arbitrary scale SR can be utilized to identify the details of an object in a satellite image, as shown in Figure 1. The arbitrary-scale SR fulfills the common requirement of arbitrarily zoom in on an image. Examples like this prove that an arbitrary-scale SR is essential. Besides CiaoSR [5], pioneering work in arbitrary-scale SR, various methods [6–8] have been proposed. However, these methods suffer from long training times because they need to train different scales of LR images and memory problems to generate several LR images. This paper proposes a network that performs arbitrary-scale SR by training in the discrete cosine transform (DCT) [9] domain of minimum and maximum scales using conventional integer-scale weights to solve these problems. Unlike existing methods, our proposed network trains directly on the DCT domain, which allows for better high-frequency reconstruction. The proposed network extends an integer-scaled image to an arbitrary-scale target in the DCT domain. It also addresses the limitations that existing integer-scale SR models only perform at a fixed integer scale. When the image is extended, high-frequency components become scarce as the arbitrary-scale increases. Thus, we use depthwise multi-head attention and a depthwise feed-forward network to restore components, which learn directly from the DCT domain. The advantages include convergence over fewer training epochs and sparser weight matrices more conducive to reduced computation [10]. We also adjust the high-frequency restoration weight of depthwise multi head attention as a coefficient for each scale so that a single model can efficiently handle arbitrary-scale SR. This paper demonstrates performance through experiments with existing state-of-the-art models. We also demonstrate the network flexibility through experiments to integrate it with existing integer-scale models. The contributions are summarized as follows:

- We propose m-DCTformer, a transformer network structure with direct training in the DCT domain for arbitrary-scale SR. Unlike traditional arbitrary-scale models, training from DCT components can improve reconstruction performance by focusing on high-frequency information and converge over fewer training epochs and computations.
- The m-DCTformer inserts a weight-modulation layer into the network trained at the minimum scale to modulate the existing weights up to the maximum scale. The weights handle arbitrary-scales by modulating the amount depending on the coefficient value, solving computational and memory problems that traditional arbitrary-scale SR models have when training contiguous LR images.
- The m-DCTformer demonstrates its flexibility through integration with the existing integer-scale SR model, and its applicability in real-world scenarios is verified through experiments.

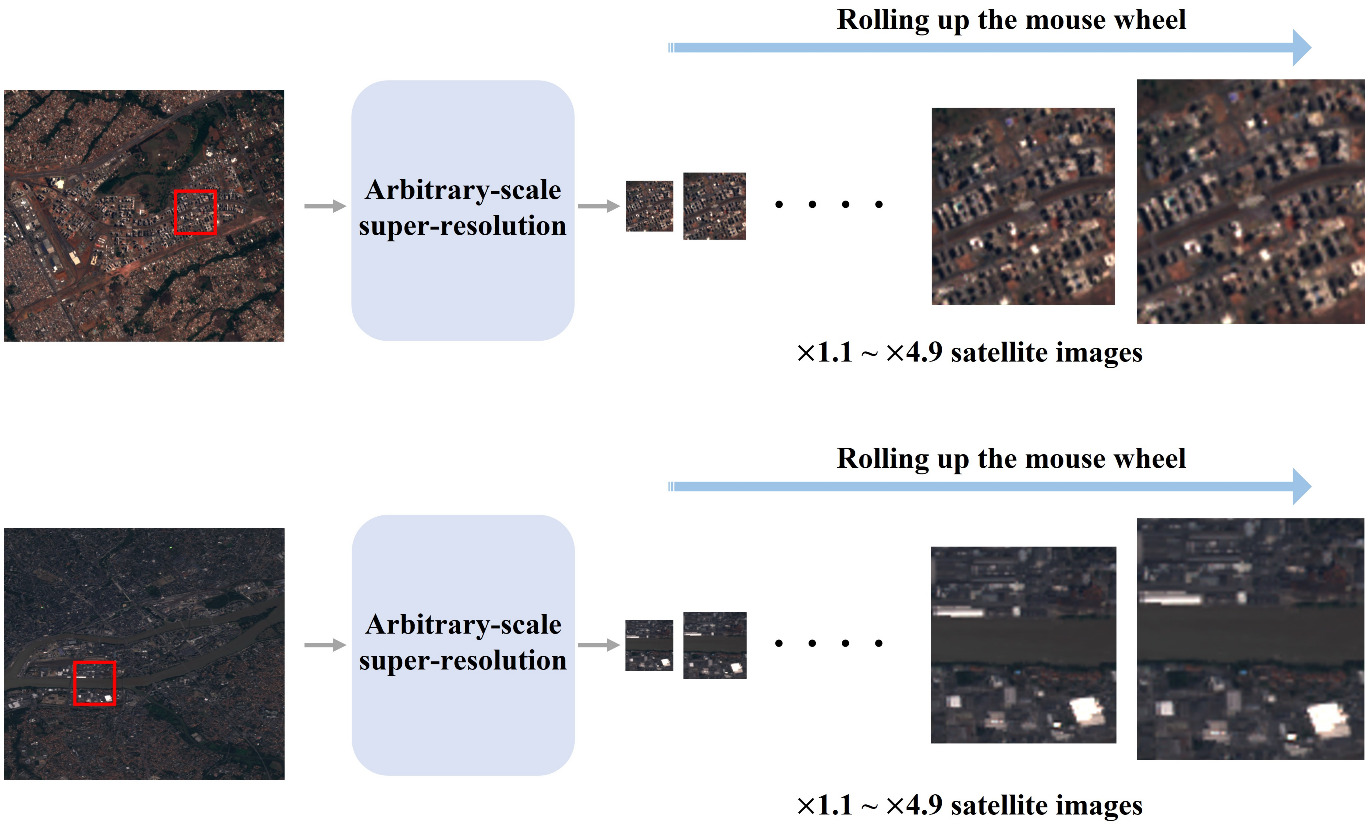
Rolling up the mouse wheel
Arbitrary-scale super-resolution
×1.1 ~ ×4.9 satellite images
Rolling up the mouse wheel
Arbitrary-scale super-resolution
×1.1 ~ ×4.9 satellite images

Figure 1. Example application of arbitrary-scale super-resolution in satellite image processing task.

2. Related Work

2.1. Single Image Super-Resoultion

The SISR technique [11–15] is well known in computer vision and aims to generate an HR image from a single LR counterpart. Early deep learning models, such as the SR convolutional neural network (SRCNN) [16] and fast SR CNN [17], use shallow architectures to learn mappings from LR to HR images. Very deep SR [18] and enhanced deep SR (EDSR) [19] attempt to increase model depth further using residual connections, allowing them to learn more complex mappings and improve reconstruction quality. The efficient sub pixel CNN [20] introduced an approach to learning an upsampling filter array in LR space, extracting feature maps from LR images and upsampling them to produce HR output. In addition, the residual dense network [21] uses a residually dense structure to learn hierarchical representations from all feature maps. Moreover, the storage area network [22] employs feature maps of input images to model relationships between neighboring pixels and dynamically highlights important features. Super-resolution [23] using normalizing flow, called SRFlow, models the conditional probability distribution between HR and LR images, facilitating accurate transformation. Swin-infrared (IR) [24] is a new SISR model that integrates the Swin transformer, a hierarchical transformer with representations computed using shifted windows, into image restoration tasks, including SISR. This model displays remarkable performance in various SR tasks. However, these methods share a fundamental limitation in that they are specifically trained and optimized for certain integer scales, rendering them ineffective in handling noninteger or arbitrary-scale SR scenarios. This research addresses this limitation by introducing a weight-modulation mechanism tailored to the DCT domain that can effectively adjust weights for high-frequency restoration according to the desired scale. Compared to these integer SISR methods, our method handles arbitrary-scale SR, making it more applicable to real-world scenarios such as Figure 1. Furthermore, the modulation is performed by a scale-dependent coefficient, allowing the model to accommodate integer and noninteger scales, providing a more comprehensive solution for arbitrary-scale SR. This advance overcomes the scale limitation of Swin-IR, increasing the flexibility and applicability of SR models.

2.2. Arbitrary Scale Super-Resolution

The arbitrary-scale SISR aims to enhance flexibility by accommodating integer- and noninteger-scale factors, overcoming the shortcomings of conventional SISR. Meta-SR [25] advances SR models by facilitating arbitrary-scale SR using a meta-upscale module. In addition, SRWarp [26] introduces a blend of warping and SR techniques, delivering an adaptive warping layer for resampling kernel prediction and multiscale blending for richer information extraction from the input. However, in SRWarp, replacing the upscale module with integer-scale SR models led to a drop in performance. ArbSR [27] employs a plug-in module with dynamic scale-aware filters, offering effective management of various scale factors but struggling with integer-scale factors. In addition, LTE [28] eemphasizes high-frequency details for arbitrary-scale SISR by estimating dominant frequencies and Fourier coefficients but tends to favor learning low-frequency components, which might prevent it from capturing minute high-frequency details. Further, CiaoSR introduced a continuous implicit attention-in-attention network, promoting the adaptive aggregation of local features, but its reliance on attention mechanisms might not be universally effective across all SR scenarios. This paper proposes a weight-modulation mechanism in the DCT domain to address these challenges. Because our proposed mechanism is directly trained by the DCT coefficient, it can restore better high-frequency details. Moreover, our weight modulation addresses model capacity limitations compared to traditional arbitrary-scale SR models. The model aims to offer a memory-efficient model capable of delivering a superior performance in integer and non integer scale factors without sacrificing the quality of the SR images.

3. Method

This section describes the proposed m-DCTformer framework. In Figure 2, the m-DCTformer applies integer-scale SR to an LR image and proceeds with the SR to the target decimal scale. First, we present an overview of the proposed framework, followed by the detailed implementation of modules.

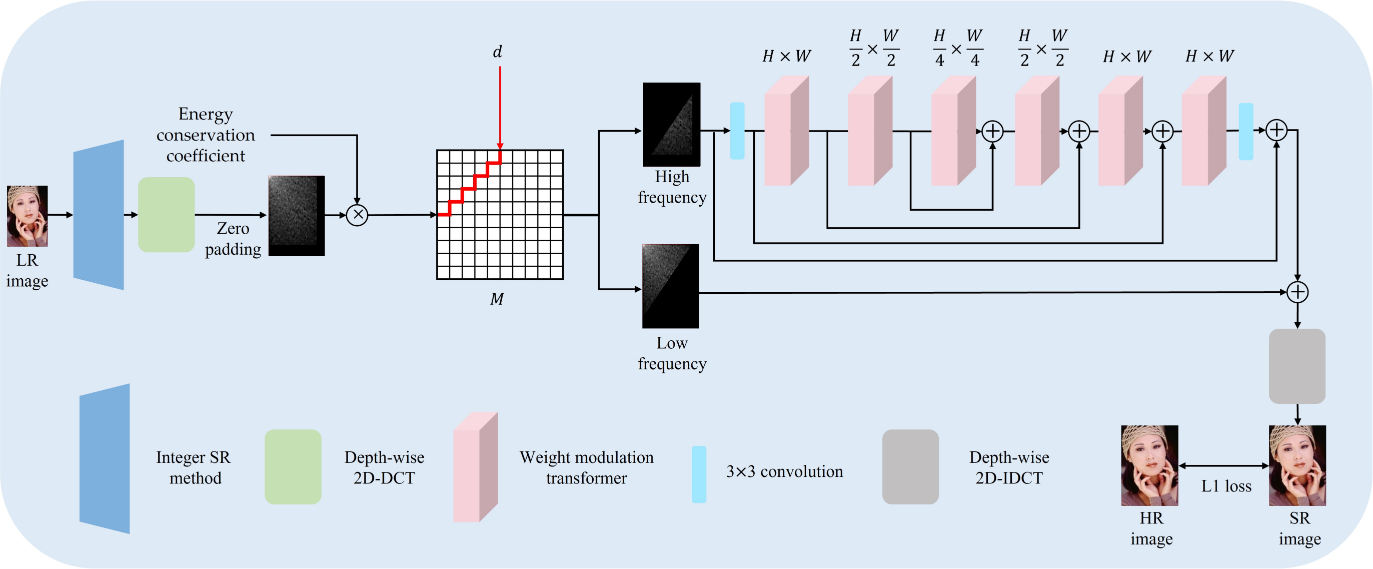

Figure 2. Architecture of the m-DCTformer.

3.1. Overview

The main goal is to proceed with arbitrary-scale SR up to the target scale based on the results of integer-scale SR. Figure 2 illustrates a comprehensive block diagram of the proposed m-DCTformer. Given an LR image as input, we proceed with an integer-scale SR using the appropriate weights for each scale factor (×2, ×3 and ×4). The result of the integer-scale SR is input into a depthwise two-dimensional (2D) DCT and expanded using zero padding by the target decimal scale. The expanded DCT domain is divided into high and low frequencies. In this process, the high-frequency components are lost due to the expansion. The process to restore these components is described next.

First, the high-frequency DCT domains are embedded into low-level features using a 3 × 3 convolution. The embedded low-level feature is input into a modulation transformer block as an encoder-decoder. Second, the input low-level features are subjected to downsampling and upsampling. The upsampling and downsampling methods are the pixel-shuffle and pixel-unshuffle, respectively. In addition, skip connections are used to restore high-frequency components. The downsampling and upsampling processes with skip connections preserve the fine structural characteristics of restored high-frequency DCT detail components. Furthermore, the original high-frequency DCT components are added to the restored high-frequency component resulting from the last 3 × 3 convolution layer. Finally, the restored high-frequency component is combined with a low-frequency component and is transformed into the spatial domain using the depthwise 2D inverse DCT (IDCT). The integer-scale SR is frozen, and the transformed spatial domain image trains the network with L1 loss with the SR image and ground truth (GT) image. In summary, the main points of m-DCTformer are as follows:

First, using the results of integer-scale SR, m-DCTformer extracts the high-frequency components using depthwise 2D DCT. This process is described in detail in Section 3.3.1. Second, the extracted high-frequency components are trained with depthwise multi-head attention and depthwise feed-forward network to restore the lost high-frequency components. This process is described in detail in Section 3.3.2. Third, we train weight modulation at the maximum scale factor and modulate to the target arbitrary-scale. This process is described in detail in Section 3.3.3.

3.2. *Depthwise 2D DCT*

A spatial domain image can be transformed into a spectral domain image. This paper uses the DCT, which decomposes the image into a cosine function and produces only real values for the spectral representation. A discrete image of size $N \times M$ input in the 2D spatial domain can be represented by a DCT in the frequency domain as follows:

$$F(u,v) = \alpha(u)\beta(v) \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x,y)\gamma(x,y,u,v), \tag{1}$$

$$\gamma(x,y,u,v) = \cos\left(\frac{\pi(2x+1)u}{2N}\right) \cos\left(\frac{\pi(2y+1)v}{2M}\right), \tag{2}$$

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & u = 0 \\ \sqrt{\frac{2}{N}} & u \neq 0 \end{cases}, \tag{3}$$

$$\beta(v) = \begin{cases} \sqrt{\frac{1}{M}} & v = 0 \\ \sqrt{\frac{2}{M}} & v \neq 0 \end{cases}, \tag{4}$$

$$f(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} \alpha(u)\beta(v)F(u,v)\gamma(x,y,u,v). \tag{5}$$

In Equation (1), $f(x,y)$ is the pixel value at position (x,y) in the input image, and $F(u,v)$ represents the DCT coefficient value at position (u,v). It also represents the depthwise 2D DCT in Figure 2. Equations (2)–(4) define the cosine basis function and regularization constants. Conversely, an image transformed into the frequency domain can be transformed into the spatial domain using the 2D IDCT, as presented in Equation (5). This process is depicted as a depthwise 2D IDCT in Figure 2. The high-frequency mask divides the transformed DCT into high and low frequencies. Mask M is expressed as follows:

$$M(u,v) = \begin{cases} 0 & D(u,v) \leq d \\ 1 & otherwise \end{cases}, \tag{6}$$

where D denotes the index of the zig-zag scanning in Figure 2, and d denotes the parameter to extract high frequency components. High-frequency DCT components are extended from an integer-scale factor to the target arbitrary-scale factor; thus, the high-frequency component is lacking. The energy conservation coefficient allows for restoring the image brightness by multiplying the value when the image is expanded in the DCT domain. However, high-frequency components are still lacking. We restored the DCT domain with the lacking high-frequency component with several weight-modulation transformer blocks to solve this.

3.3. Weight Modulation Tansformer

3.3.1. Depthwise Multi-Head Attention

We note that depthwise multi-head attention receives as input the high-frequency components obtained from depthwise 2D DCT. In addition, the obtained high-frequency components are extracted as Q, K, and V using depthwise convolution. Meanwhile, depthwise multi-head attention can apply self-attention [29] across channels, repeated h times independently, depending on hyperparameter h, as depicted in Figure 3. Another key point is using depthwise convolutions to emphasize the local context. A normalized tensor layer is input and generates a query, a key projection enriched with the local context. Then a 1 × 1 convolution is applied to aggregate the pixelwise cross-channel context, and a 3 × 3 depthwise convolution encodes the channelwise spatial context. The convolutional layers of the depthwise multi-head attention network are bias-free. Next, we reshaped the query and key projections such that their dot-product interaction generates a transposed attention map A. The depthwise multi-head attention can be expressed as follows:

$$F_a = Attention(Q,K,V) + F_0, \tag{7}$$

$$Attention(Q,K,V) = V \cdot softmax(K \cdot Q/\epsilon), \tag{8}$$

where the input and output feature maps are F_a and F_0. Moreover, $\cdot$ denotes the dot product. In Figure 3, the first weight-modulation transformer block receives the feature F_0 extracted through a 3 × 3 convolution layer as input. The Q, K and V matrices are obtained after the input tensor is reshaped. The learnable ϵ value can be obtained to control the size of the dot product of K and Q before applying the softmax function. After applying softmax to the attention map obtained by Q and K, we multiplied it by V and added the resulting value to F_0 to obtain F_a. Since depthwise convolution is involved in obtaining Q, K, and V, we get an attention map that is more relevant to the high-frequency components lost during the training process. F_a is restored the depthwise feed-forward network.

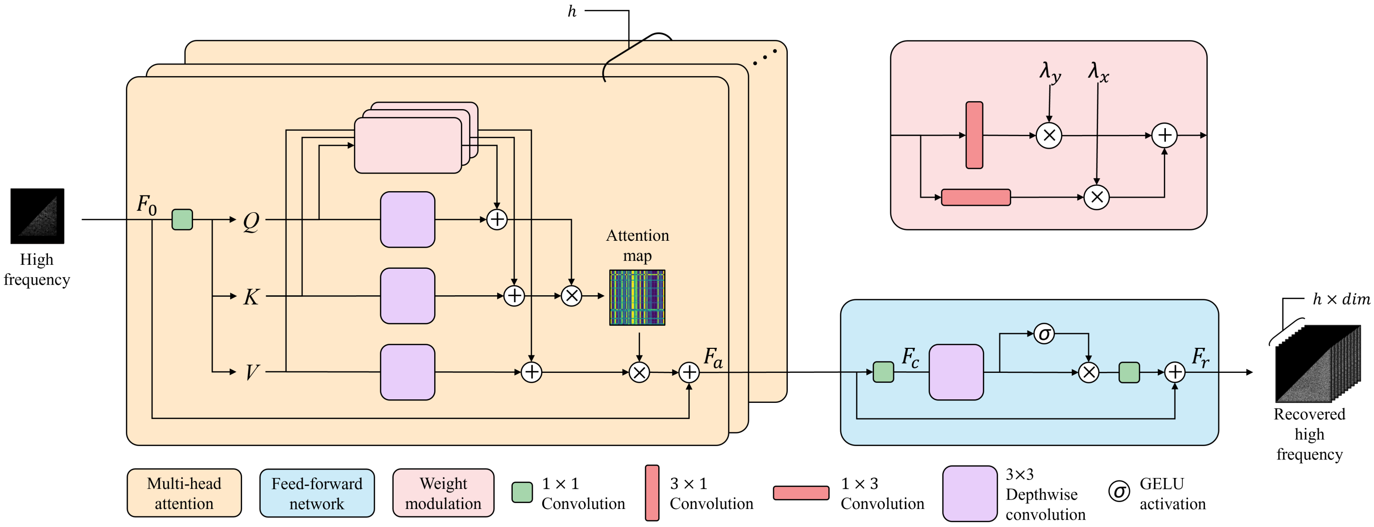

Figure 3. Architecture of weight modulation transformer.

3.3.2. Depthwise Feed-Forward Network

The regular feed-forward network operates on each pixel location separately and identically. However, we adopted the depthwise feed-forward network in Figure 3. As in the regular feed-forward network, two 1 × 1 convolutions are performed. The input layer 1 × 1 convolution expands the feature channels, and the output layer 1 × 1 convolution reduces them back to the original input dimension. Then, we extract the features using a 3 × 3 depthwise convolution, allowing the extraction of spatial information from neighboring pixel positions, which can be used to learn local features for effective reconstruction. A gate mechanism forms an elementwise multiplication of two parallel paths output from the depthwise convolution. Prior to this multiplication, we applied the Gaussian error linear unit (GELU) [30] activation function to one of the parallel paths, introducing nonlinearity into the model. This activation function allows the model to learn and adapt to complex data patterns. The gate mechanism contributes to the robustness of the model, enhancing its ability to adapt to different image patterns. Given an attentional tensor F_a, a depthwise feed-forward network can be expressed as follows:

$$F_r = F_a + Gate(F_c), \tag{9}$$

$$F_c = Conv_{1\times1}(F_a), \tag{10}$$

$$Gate(F_c) = Conv_{1\times1}(\sigma(Conv_d(F_c)) \odot Conv_d(F_c), \tag{11}$$

where $\odot$ denotes element-wise multiplication, σ denotes the GELU activation function, $Conv_{1\times1}$ represents the 1 × 1 convolution layer, and $Conv_d$ indicates the depthwise convolution layer. The depthwise feed-forward network enables effective restoration of the high-frequency components through depthwise convolution. By incorporating 3 × 3 depthwise convolution and the GELU activation function into a gate mechanism, the depthwise feed-forward network provides a more refined and context-aware approach to feature transformation. Furthermore, F_r, which is extracted as a result of Equation (9), is the recovered high frequency. In this case, the number of dimensions is the product of the hyperparameter h and the dimension of F_0, the input feature of depthwise multi-head attention. This process leads to improved representational learning capabilities and, consequently, better performance in tasks such as high frequency restoration.

3.3.3. Weight Modulation

The weight of each Q, K, and V can be modulated to $w(s_{x,max}, s_{y,max})$ using each weight modulation and an appropriate coefficient, λ_x and λ_y, which are the scale coefficients of horizontal modulation filter m_x and vertical modulation filter m_y, respectively. The weight modulation can be formulated as follows:

$$w(s_x, s_y) = w(s_{x,max}, s_{y,max}) * \lambda_x m_x * \lambda_y m_y, \tag{12}$$

where $*$ denotes a convolution operation. Each weight starts at the maximum scale factor $s_{x,max}$, $s_{y,max}$ and modulates to the target arbitrary-scale factor s_x, s_y. Figure 4 depicts the process of the weight-modulation layer. We designed them separately as 3 × 1 and 1 × 3 convolution layers to process in each horizontal and vertical direction, respectively. Each layer trains independently at the minimum arbitrary-scale factors of $s_{x,min}$ and $s_{y,min}$. Therefore, we modulate the weights w with coefficients λ_x and λ_y according to the target arbitrary-scale factor. The modulated feature f and the coefficient λ_x and λ_y to use weight modulation can be expressed as follows:

$$f_y = (f * w) * \lambda_y w_y = f * (w * \lambda_y w_y), \tag{13}$$

$$f_x = (f * w) * \lambda_x w_x = f * (w * \lambda_x w_x), \tag{14}$$

$$\lambda_x = (s_{x,max} - s_x)/(s_{x,max} - s_{x,min}), \tag{15}$$

$$\lambda_y = (s_{y,max} - s_y)/(s_{y,max} - s_{y,min}). \tag{16}$$

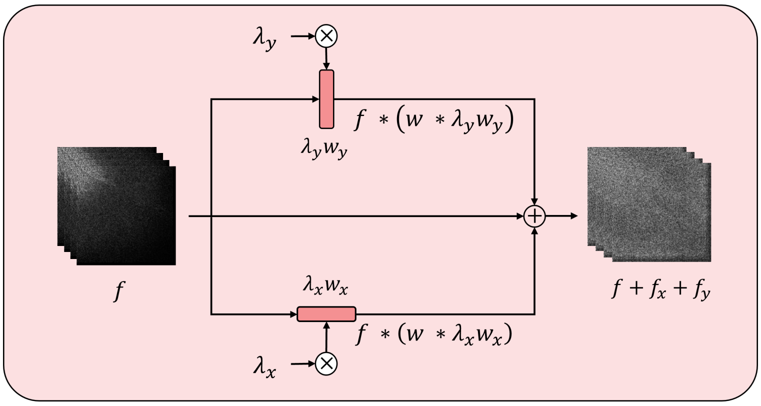

Figure 4. Weight modulation process.

4. Experiment

This section presents the experimental results and discusses their implications. We introduce the experimental setup and evaluate m-DCTformer on the datasets for training and valuation. Finally, we analyze the results of the experiments comparing m-DCTformer with other models.

4.1. Experimental Setup

During training, we input a batch of LR images into the framework following previous work LTE. The corresponding LR images were cropped into 64 × 64 patches. The patches were augmented by a random horizontal flip, vertical flip, and 90° rotation. We set the batch size to 4 and used the Adam optimizer [31] for training, with L1 loss instead of the mean squared loss (MSE) or L2 loss. The m-DCTformer has six weight modulation transformer blocks. The number of dimensions in each of the six weight modulation transformer blocks is 48, 96, 192, 96, 48, and 48. We train m-DCTformer for 1000 epochs, initializing the learning rate to 1×10^{-4} and decaying by 0.5 at epochs [200, 400, 600, 800]. We set the hyperparameter d to 20 and the h values to [4, 6, 6]. Moreover, we set the weighting for the gradient moving average to 0.9 and the weighting for the squared gradient moving average to 0.999 in the Adam optimizer [31]. Furthermore, we used an NVIDIA RTX 3090 24 GB for training. The coefficient of each weight modulation used in the test was calculated as the ratio of the maximum arbitrary-scale to the minimum arbitrary-scale. We used the classical method EDSR and the state-of-the-art method hybrid attention transformer (HAT) [32] for integer-scale SR.

4.2. Dataset

We used the DIV2K dataset [33] for CiaoSR, LTE, and our m-DCTformer training. It consists of 1000 images in 2K resolutions and provides low-resolution counterparts with down-sampling scales, ×2, ×3, and ×4, generated by the bicubic interpolation method. We also evaluate the performance on the validation set of Set5 [34], Set14 [35], Urban100 [36] and real-world SR dataset [37] in terms of peak signal-to-noise (PSNR) and in terms of structural similarity index measure (SSIM) values.

4.3. Quantitative and Qualitative Results on Set5, Set14, Urban 100 Datasets

This section compares the performance of the proposed m-DCTformer with the state-of-the-art arbitrary-scale SR methods CiaoSR and LTE. Each arbitrary-scale method is trained on the LR bicubic DIV2K dataset. For the proposed model, we only trained with LR datasets with scale factors of ×2.1 and ×4.9, whereas for the other models, we used scale factors of ×2, ×3, and ×4. We also evaluated the proposed model quantitatively using the PSNR and SSIM. Tables 1–3 show the quantitative results with other approaches and our m-DCTformer. Moreover, they use EDSR and HAT in integer SR, and they are obtained from Set5, Set14, and Urban100 datasets. Table 1 shows our EDSR + m-DCTformer outperforms EDSR + CiaoSR by an average of 0.23 dB and 0.0462 dB and outperforms EDSR + LTE by an average of 0.17 dB and 0.0045 dB in terms of PSNR and SSIM, respectively. Our HAT + m-DCTformer outperforms HAT + CiaoSR by an average of 0.71 dB and 0.0116 dB and outperforms HAT + LTE by an average of 0.43 dB and 0.0076 dB in terms of PSNR and SSIM, respectively. Table 2 shows our EDSR + m-DCTformer outperforms EDSR + CiaoSR by an average of 0.15 dB and 0.0449 dB and outperforms EDSR + LTE by an average of 0.1 dB and 0.0037 dB in terms of PSNR and SSIM, respectively. Our HAT + m-DCTformer outperforms HAT + CiaoSR by an average of 0.24 dB and 0.0043 dB and outperforms HAT + LTE by an average of 0.51 dB and 0.0098 dB in terms of PSNR and SSIM, respectively. Table 3 shows our EDSR + m-DCTformer outperforms EDSR + CiaoSR by an average of 0.64 dB and 0.0055 dB and outperforms EDSR + LTE by an average of 0.22 dB and 0.0061 dB in terms of PSNR and SSIM, respectively. Our HAT + m-DCTformer outperforms HAT + CiaoSR by an average of 0.72 dB and 0.0048 dB and outperforms HAT + LTE by an average of 1.02 dB and 0.0078 dB in terms of PSNR and SSIM, respectively. Compared to other methods, the model demonstrates high PSNR performance, especially in Urban 100. As shown in Figures 5–17, m-DCTformer exhibits the closest results to the GT image compared to the other models. Figures 5 and 6 show the qualitative results of m-DCTformer and other models based on EDSR in integer SR, obtained from the Set5 dataset. Figures 7 and 8 show the qualitative results of m-DCTformer and other models based on EDSR in integer SR, obtained from the Set 14 dataset. Figures 9–14 show the qualitative results of m-DCTformer and other models based on EDSR in integer SR, obtained from the Urban 100 dataset. Figures 15–17 show the qualitative results of m-DCTformer and other models based on HAT in integer SR, obtained from the Set 14 and Urban 100 dataset. In particular, it recovers handrails, building lines, and other aspects very well in the Urban 100 datasets. This outcome is because the depthwise 2D DCT module converts to the DCT domain and extracts high frequencies to restore damaged high frequencies and because the weight modulation is divided into horizontal and vertical directions to modulate the weight of the existing maximum noninteger-scale factor.

Table 1. Quantitative results with other approaches and our m-DCTformer with EDSR and HAT on Set5 datasets. Bold represents the best peak signal-to-nose ratio (PSNR) and similarity index measure (SSIM) scores.

Dataset	**Set5**					
Method	EDSR [19] + CiaoSR [5]		EDSR [19] + LTE [28]		EDSR [19] + ours	
Metric	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
2.1	37.35	0.9313	37.47	0.9559	**37.51**	**0.9563**
2.2	36.88	0.9265	37.05	0.9527	**37.12**	**0.9533**
2.3	36.58	0.9220	36.69	0.9494	**36.77**	**0.9498**
2.4	36.18	0.9176	36.30	0.9458	**36.36**	**0.9462**
2.5	35.86	0.9131	35.93	0.9425	**35.99**	**0.9426**
2.6	35.54	0.9086	35.65	**0.9394**	**35.67**	0.9393
2.7	35.23	0.9042	35.28	**0.9362**	**35.31**	0.9360
2.8	34.92	0.8996	34.97	0.9329	**35.01**	**0.9326**
2.9	34.60	0.8956	**34.75**	**0.9298**	34.74	0.9293
3.1	34.13	0.8871	34.19	0.9228	**34.40**	**0.9257**
3.2	33.92	0.8834	34.03	0.9200	**34.22**	**0.9231**
3.3	33.65	0.8786	33.76	0.9166	**33.98**	**0.9201**
3.4	33.45	0.8742	33.53	0.9131	**33.72**	**0.9169**
3.5	33.23	0.8708	33.30	0.9104	**33.47**	**0.9138**
3.6	32.95	0.8667	32.98	0.9054	**33.12**	**0.9090**
3.7	32.75	0.8628	32.73	0.9018	**32.93**	**0.9064**
3.8	32.53	0.5857	32.58	0.8989	**32.73**	**0.9032**
3.9	32.35	0.8546	32.37	0.8945	**32.51**	**0.8993**
4.1	31.99	0.8474	31.98	0.8875	**32.17**	**0.8942**
4.2	31.83	0.8440	31.84	0.8843	**32.10**	**0.8914**
4.3	31.62	0.8394	31.60	0.8801	**31.85**	**0.8878**
4.4	31.38	0.8356	31.43	0.8769	**31.71**	**0.8853**
4.5	31.24	0.8328	31.23	0.8722	**31.56**	**0.8825**
4.6	30.98	0.8279	31.08	0.8684	**31.43**	**0.8797**
4.7	30.85	0.8251	30.88	0.8645	**31.21**	**0.8761**
4.8	30.70	0.8221	30.75	0.8623	**31.03**	**0.8733**
4.9	30.53	0.8166	30.50	0.8565	**30.84**	**0.8690**
Method	HAT [32] + CiaoSR [5]		HAT [32] + LTE [28]		HAT [32] + ours	
Metric	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
2.1	37.46	0.9522	37.67	0.9567	**37.94**	**0.9587**
2.2	36.97	0.9472	37.27	0.9537	**37.52**	**0.9556**
2.3	36.67	0.9429	36.90	0.9505	**37.14**	**0.9522**
2.4	36.24	0.9414	36.51	0.9471	**36.71**	**0.9488**
2.5	35.94	0.9401	36.17	0.9441	**36.27**	**0.9452**
2.6	35.60	0.9396	35.87	0.9409	**35.96**	**0.9419**
2.7	35.31	0.9351	35.51	0.9377	**35.55**	**0.9382**
2.8	35.00	0.9305	**35.26**	0.9348	35.24	**0.9349**
2.9	34.71	0.9268	35.00	0.9319	**35.02**	**0.9319**
3.1	34.22	0.9218	34.50	0.9255	**35.00**	**0.9305**
3.2	33.99	0.9200	34.32	0.9226	**34.80**	**0.9278**
3.3	33.72	0.9151	34.07	0.9197	**34.52**	**0.9250**
3.4	33.56	0.9119	33.86	0.9164	**34.31**	**0.9226**
3.5	33.30	0.9084	33.60	0.9132	**34.13**	**0.9201**
3.6	33.01	0.9059	33.25	0.9085	**33.83**	**0.9162**
3.7	32.80	0.9018	33.08	0.9055	**33.57**	**0.9128**
3.8	32.60	0.8984	32.89	0.9025	**33.44**	**0.9101**
3.9	32.39	0.8942	32.64	0.8975	**33.13**	**0.9065**
4.1	32.01	0.8899	32.28	0.8919	**32.92**	**0.9032**
4.2	31.87	0.8843	32.22	0.8889	**32.78**	**0.9009**
4.3	31.71	0.8807	32.00	0.8852	**32.61**	**0.8977**
4.4	31.50	0.8769	31.78	0.8812	**32.50**	**0.8963**
4.5	31.34	0.8726	31.62	0.8778	**32.29**	**0.8929**
4.6	31.10	0.8701	31.44	0.8739	**32.11**	**0.8894**
4.7	30.97	0.8676	31.28	0.8703	**31.99**	**0.8873**
4.8	30.79	0.8659	31.10	0.8680	**31.73**	**0.8846**
4.9	30.67	0.8601	30.90	0.8627	**31.62**	**0.8819**

Table 2. Quantitative results with other approaches and our m-DCTformer with EDSR and HAT on Set14 datasets. Bold represents the best peak signal-to-nose ratio (PSNR) and similarity index measure (SSIM) scores.

Dataset	**Set14**					
Method	EDSR [19] + CiaoSR [5]		EDSR [19] + LTE [28]		EDSR [19] + ours	
Metric	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
2.1	32.91	0.8801	33.19	0.9117	**33.30**	**0.9123**
2.2	32.52	0.8690	32.70	0.9029	**32.83**	**0.9036**
2.3	32.13	0.8594	32.31	0.8955	**32.41**	**0.8960**
2.4	31.81	0.8510	31.99	**0.8888**	**32.08**	0.8885
2.5	31.50	0.8418	31.63	**0.8807**	**31.71**	0.8803
2.6	31.19	0.8336	31.33	**0.8727**	**31.40**	0.8720
2.7	30.92	0.8252	31.05	**0.8656**	**31.08**	0.8645
2.8	30.73	0.8188	30.78	**0.8583**	**30.79**	0.8569
2.9	30.43	0.8093	30.50	**0.8511**	**30.51**	0.8495
3.1	30.13	0.7958	30.12	0.8374	**30.25**	**0.8387**
3.2	29.88	0.7878	29.94	0.8310	**30.02**	**0.8323**
3.3	29.70	0.7811	29.73	0.8239	**29.83**	**0.8258**
3.4	29.53	0.7742	29.54	0.8173	**29.65**	**0.8196**
3.5	29.37	0.7678	29.39	0.8110	**29.49**	**0.8134**
3.6	29.20	0.7614	29.24	0.8058	**29.33**	**0.8085**
3.7	29.05	0.7550	29.03	0.7988	**29.10**	**0.8014**
3.8	28.89	0.7497	28.86	0.7925	**28.95**	**0.7956**
3.9	28.75	0.7431	28.76	0.7879	**28.83**	**0.7914**
4.1	28.48	0.7309	28.48	0.7755	**28.61**	**0.7816**
4.2	28.35	0.7259	28.36	0.7701	**28.51**	**0.7771**
4.3	28.23	0.7209	28.19	0.7637	**28.41**	**0.7720**
4.4	28.10	0.7154	28.10	0.7583	**28.29**	**0.7677**
4.5	27.96	0.7105	27.95	0.7530	**28.11**	**0.7621**
4.6	27.84	0.7052	27.81	0.7474	**27.93**	**0.7570**
4.7	27.73	0.6998	27.72	0.7426	**27.85**	**0.7529**
4.8	27.60	0.6944	27.62	0.7377	**27.71**	**0.7480**
4.9	27.49	0.6910	27.46	0.7309	**27.59**	**0.7424**
Method	HAT [32] + CiaoSR [5]		HAT [32] + LTE [28]		HAT [32] + ours	
Metric	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
2.1	33.73	0.9179	33.57	0.9148	**34.16**	**0.9200**
2.2	33.27	0.9108	33.01	0.9064	**33.62**	**0.9114**
2.3	32.90	0.9014	32.64	0.8995	**33.21**	**0.9044**
2.4	32.56	0.8956	32.30	0.8925	**32.81**	**0.8975**
2.5	32.22	0.8873	31.88	0.8839	**32.41**	**0.8894**
2.6	31.89	0.8808	31.58	0.8760	**32.03**	**0.8812**
2.7	31.61	0.8729	31.26	0.8689	**31.66**	**0.8732**
2.8	31.31	0.8643	30.99	0.8619	**31.34**	**0.8655**
2.9	31.03	0.8576	30.72	0.8551	**31.04**	**0.8577**
3.1	30.95	0.8490	30.30	0.8415	**31.01**	**0.8496**
3.2	30.38	0.8409	30.11	0.8353	**30.74**	**0.8437**
3.3	30.22	0.8335	29.91	0.8283	**30.52**	**0.8368**
3.4	30.03	0.8282	29.75	0.8220	**30.29**	**0.8304**
3.5	29.84	0.8200	29.58	0.8157	**30.11**	**0.8243**
3.6	29.68	0.8156	29.41	0.8104	**29.94**	**0.8197**
3.7	29.69	0.8088	29.21	0.8031	**29.69**	**0.8134**
3.8	29.35	0.8042	29.03	0.7973	**29.52**	**0.8074**
3.9	29.20	0.7998	28.95	0.7929	**29.41**	**0.8030**
4.1	28.88	0.7901	28.70	0.7816	**29.14**	**0.7941**
4.2	28.73	0.7825	28.59	0.7761	**29.02**	**0.7888**
4.3	28.63	0.7780	28.45	0.7702	**28.87**	**0.7836**
4.4	28.47	0.7725	28.37	0.7656	**28.81**	**0.7798**
4.5	28.33	0.7704	28.11	0.7588	**28.67**	**0.7748**
4.6	28.22	0.7638	27.94	0.7534	**28.51**	**0.7710**
4.7	28.09	0.7595	27.86	0.7488	**28.42**	**0.7666**
4.8	27.96	0.7438	27.76	0.7437	**28.28**	**0.7624**
4.9	27.83	0.7401	27.71	0.7386	**28.17**	**0.7571**

Table 3. Quantitative results with other approaches and our m-DCTformer with EDSR and HAT on Urban100 datasets. Bold represents the best peak signal-to-nose ratio (PSNR) and similarity index measure (SSIM) scores.

Dataset	**Urban100**					
Method	EDSR [19] + CiaoSR [5]		EDSR [19] + LTE [28]		EDSR [19] + ours	
Metric	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
2.1	30.85	0.9196	31.70	0.9224	**32.00**	**0.9246**
2.2	30.25	0.9105	31.20	0.9150	**31.42**	**0.9166**
2.3	29.65	0.9002	30.74	0.9076	**30.87**	**0.9082**
2.4	29.32	0.8948	30.33	**0.9004**	**30.37**	0.8999
2.5	28.95	0.8873	**29.92**	**0.8929**	29.91	0.8917
2.6	28.65	0.8798	**29.56**	**0.8856**	29.47	0.8831
2.7	28.14	0.8649	**29.21**	**0.8783**	29.07	0.8750
2.8	28.06	0.8665	**28.89**	**0.8712**	28.70	0.8667
2.9	27.87	0.8601	**28.60**	**0.8643**	28.38	0.8588
3.1	27.16	0.8460	28.07	0.8507	**28.46**	**0.8573**
3.2	26.94	0.8380	27.83	0.8441	**28.20**	**0.8508**
3.3	26.66	0.8318	27.60	0.8372	**27.93**	**0.8437**
3.4	26.47	0.8242	27.38	0.8307	**27.69**	**0.8372**
3.5	26.24	0.8176	27.16	0.8239	**27.46**	**0.8304**
3.6	26.09	0.8120	26.96	0.8173	**27.24**	**0.8239**
3.7	25.88	0.8041	26.76	0.8107	**27.03**	**0.8172**
3.8	25.72	0.7990	26.58	0.8044	**26.82**	**0.8109**
3.9	25.55	0.7930	26.41	0.7981	**26.64**	**0.8046**
4.1	25.22	0.7806	26.07	0.7851	**26.43**	**0.7964**
4.2	25.07	0.7741	25.91	0.7788	**26.27**	**0.7907**
4.3	24.94	0.7686	25.77	0.7731	**26.12**	**0.7853**
4.4	24.78	0.7643	25.62	0.7666	**25.97**	**0.7795**
4.5	24.49	0.7494	25.49	0.7610	**25.84**	**0.7744**
4.6	24.57	0.7550	25.34	0.7544	**25.69**	**0.7685**
4.7	24.26	0.7401	25.21	0.7487	**25.55**	**0.7629**
4.8	30.70	0.8221	25.09	0.7431	**25.42**	**0.7579**
4.9	30.53	0.8166	24.96	0.7369	**25.28**	**0.7522**
Method	HAT [32] + CiaoSR [5]		HAT [32] + LTE [28]		HAT [32] + ours	
Metric	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
2.1	32.66	0.9302	32.45	0.9295	**33.32**	**0.9373**
2.2	32.10	0.9253	31.93	0.9227	**32.63**	**0.9302**
2.3	31.38	0.9138	31.47	0.9159	**32.00**	**0.9227**
2.4	31.10	0.9101	31.03	0.9092	**31.45**	**0.9150**
2.5	30.71	0.9036	30.63	0.9025	**30.92**	**0.9072**
2.6	30.35	0.8977	30.24	0.8956	**30.43**	**0.8992**
2.7	29.46	0.8842	29.89	0.8890	**29.98**	**0.8913**
2.8	**29.73**	**0.8857**	29.56	0.8822	29.57	0.8834
2.9	**29.45**	**0.8794**	29.27	0.8759	29.19	0.8756
3.1	28.98	0.8758	28.70	0.8731	**30.26**	**0.8876**
3.2	28.81	0.8701	28.46	0.877	**29.94**	**0.8821**
3.3	28.65	0.8728	28.22	0.8704	**29.61**	**0.8758**
3.4	28.51	0.8692	27.98	0.8643	**29.32**	**0.8699**
3.5	28.18	0.8601	27.76	0.8581	**29.02**	**0.8638**
3.6	27.96	0.8552	27.55	0.8520	**28.75**	**0.8578**
3.7	27.79	0.8497	27.36	0.8459	**28.49**	**0.8519**
3.8	27.53	0.8424	27.17	0.8400	**28.27**	**0.8463**
3.9	27.41	0.8374	26.99	0.8339	**28.04**	**0.8405**
4.1	27.23	0.8367	26.62	0.8319	**28.18**	**0.8404**
4.2	26.92	0.8290	26.46	0.8259	**27.98**	**0.8354**
4.3	26.75	0.8242	26.32	0.8205	**27.82**	**0.8311**
4.4	26.65	0.8191	26.16	0.8144	**27.63**	**0.8258**
4.5	26.61	0.8153	26.03	0.8093	**27.45**	**0.8211**
4.6	26.31	0.8102	25.88	0.8033	**27.30**	**0.8163**
4.7	26.12	0.8054	25.74	0.7976	**27.13**	**0.8112**
4.8	25.92	0.8007	25.62	0.7923	**26.99**	**0.8071**
4.9	25.71	0.7952	25.48	0.7864	**26.84**	**0.8021**

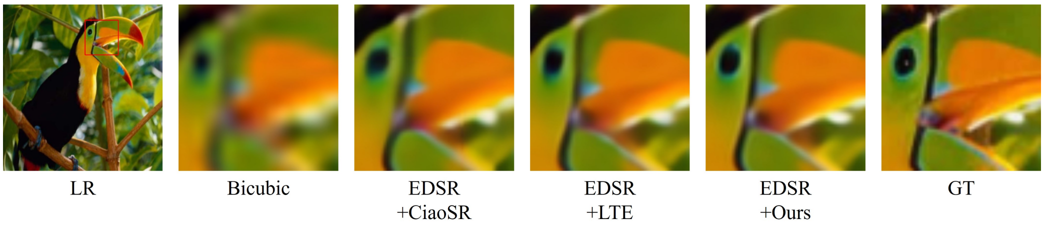

Figure 5. Qualitative comparison of the m-DCTformer with other arbitrary scale super-resolution methods for a scale factor of ×4.9 on the Set5 dataset.

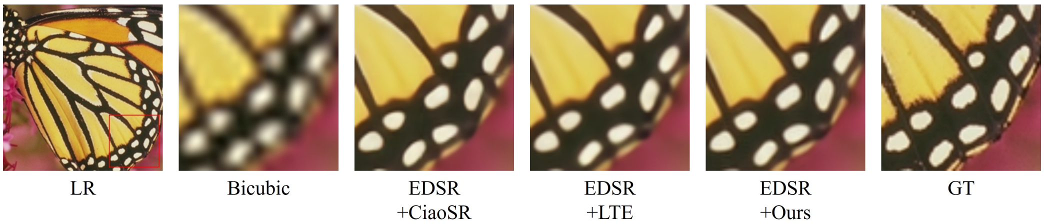

Figure 6. Qualitative comparison of the m-DCTformer with other arbitrary scale super-resolution methods for a scale factor of ×4.9 on the Set5 dataset.

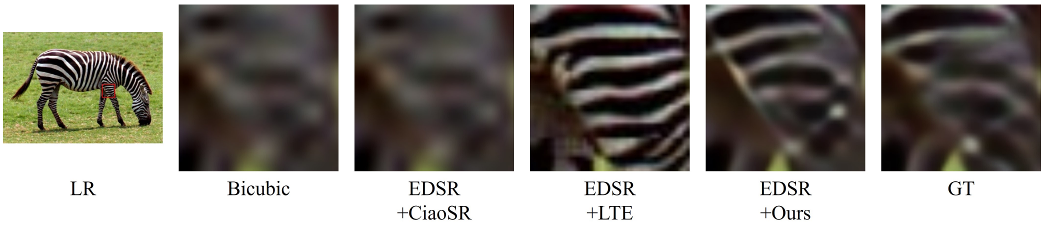

Figure 7. Qualitative comparison of the m-DCTformer with other arbitrary-scale super-resolution methods for a scale factor of ×4.9 on the Set 14 dataset.

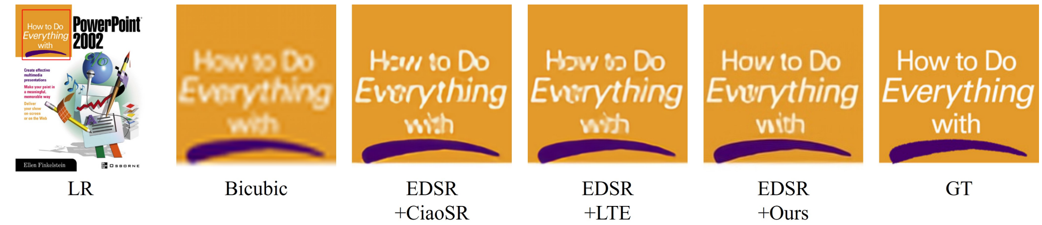

Figure 8. Qualitative comparison of the m-DCTformer with other arbitrary scale super-resolution methods for a scale factor of ×4.9 on the Set 14 dataset.

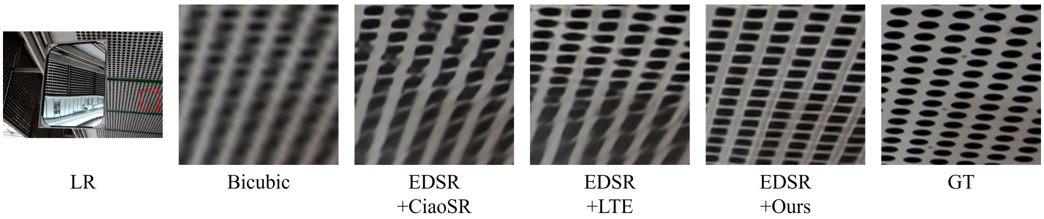

Figure 9. Qualitative comparison of the m-DCTformer with other arbitrary scale super-resolution methods for a scale factor of ×4.9 on the Urban 100 dataset.

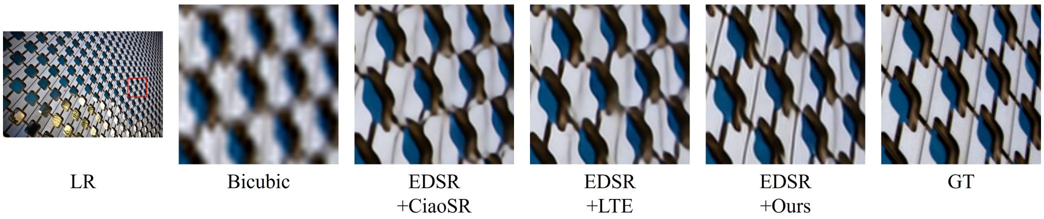

Figure 10. Qualitative comparison of the m-DCTformer with other arbitrary scale super-resolution methods for a scale factor of ×4.9 on the Urban 100 dataset.

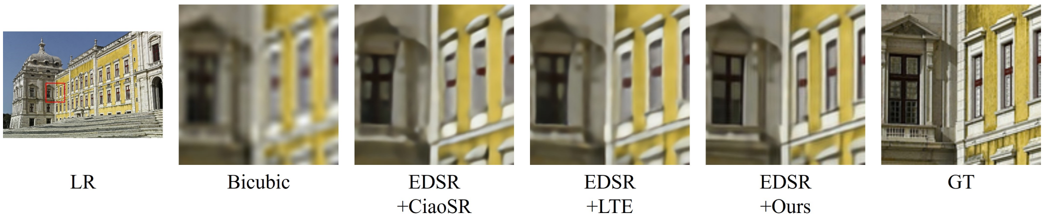

Figure 11. Qualitative comparison of the m-DCTformer with other arbitrary scale super-resolution methods for a scale factor of ×4.9 on the Urban 100 dataset.

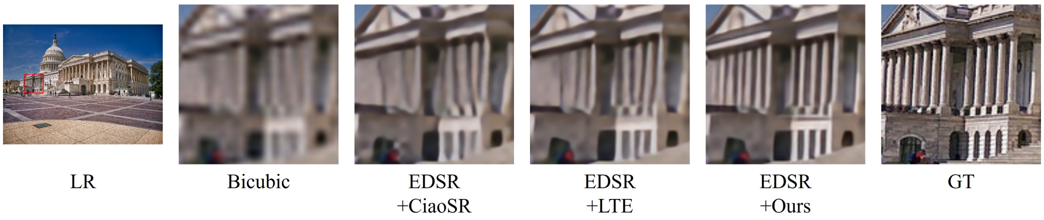

Figure 12. Qualitative comparison of the m-DCTformer with other arbitrary scale super-resolution methods for a scale factor of ×4.9 on the Urban 100 dataset.

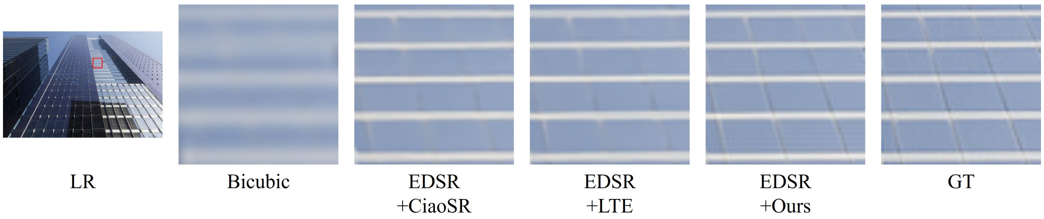

Figure 13. Qualitative comparison of the m-DCTformer with other arbitrary scale super-resolution methods for a scale factor of ×3.1 on the Urban 100 dataset.

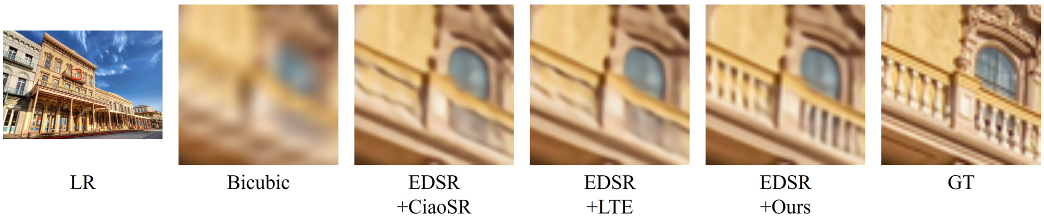

Figure 14. Qualitative comparison of the m-DCTformer with other arbitrary scale super-resolution methods for a scale factor of ×3.1 on the Urban 100 dataset.

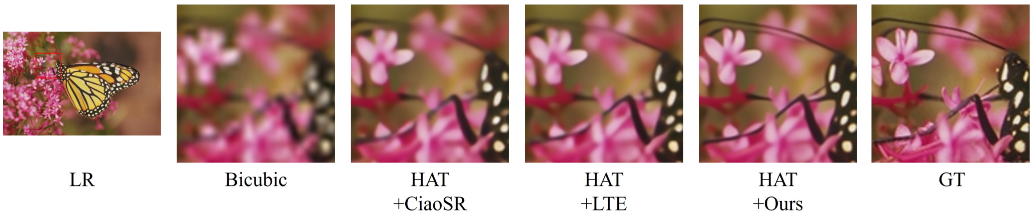

Figure 15. Qualitative comparison of the m-DCTformer with other arbitrary-scale super-resolution methods for a scale factor of ×4.9 on the Set 14 dataset.

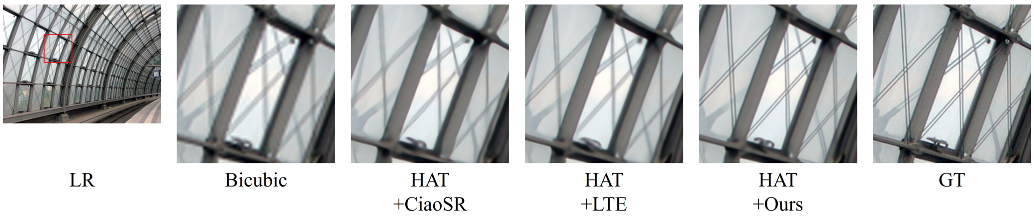

Figure 16. Qualitative comparison of the m-DCTformer with other arbitrary scale super-resolution methods for a scale factor of ×4.9 on the Urban 100 dataset.

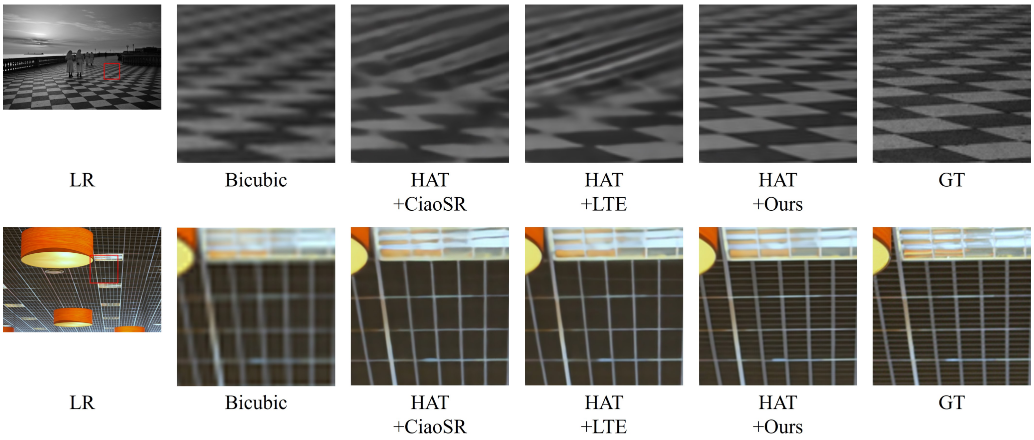

Figure 17. Qualitative comparison of m-DCTformer with other arbitrary-scale super-resolution methods for a scale factor (×4.9) on the Urban 100 datasets.

4.4. Qualitative Results on Real-World Dataset

This section applies the real-world dataset to validate a qualitative comparison with other methods. This dataset has real noise images with no GT images. Figures 18–20 show the qualitative results of m-DCTformer and other models based on HAT in integer SR, obtained from the real-world dataset for a scale factor of ×2.9. Moreover, Figures 18–20 indicate that the m-DCTformer recovers high-frequency DCT components robustly, even in a noisy real-world environment. Therefore, m-DCTformer is suitable for a real-world scenario.

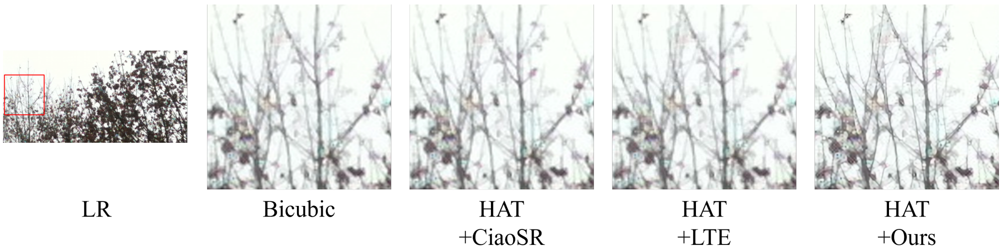

Figure 18. Qualitative comparison of the m-DCTformer with other arbitrary scale super-resolution methods for a scale factor of ×2.9 on the real-world dataset.

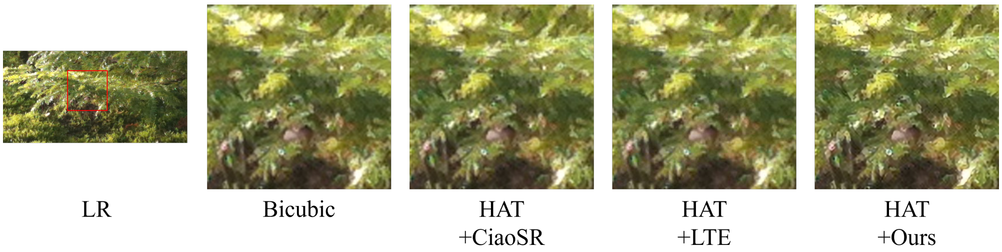

Figure 19. Qualitative comparison of the m-DCTformer with other arbitrary scale super-resolution methods for a scale factor of ×2.9 on the real-world dataset.

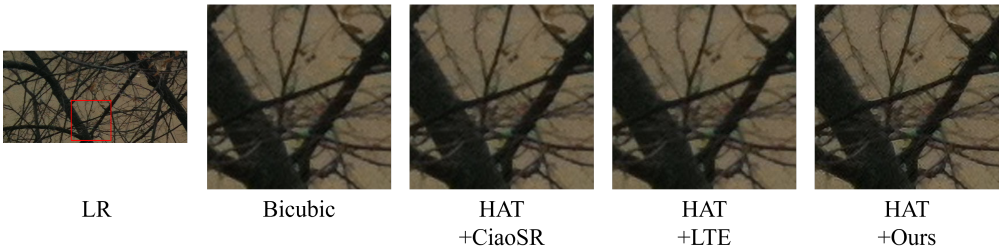

Figure 20. Qualitative comparison of the m-DCTformer with other arbitrary scale super-resolution methods for a scale factor of ×2.9 on the real-world dataset.

4.5. Complexity

Reconstructing high-quality images like DIV2K consumes considerable memory during evaluation. Table 4 compares the m-DCTformer with other arbitrary-scale SR methods for floating point operations per second (FLOPs), parameters, model capacity, and time on an Nvidia RTX 3090 24 GB environment. The FLOPs are computed as the theoretical amount of multiply-add operations in the network. Moreover, the parameters are computed from the number of parameters in a network. In addition, the model capacity denotes the capacity of the model in terms of megabytes (MB), and the time in Table 4 denotes inference time for performing a single image. The output image size is 256 × 256 and the scale factor is ×2.5. The m-DCTformer has the lowest FLOPs and fastest time because it learns high-frequency DCT components. It has the smallest memory size because it handles an arbitrary-scale from minimum to maximum with weight modulation. This efficient memory size makes it applicable to real-world scenarios.

Table 4. Comparison of the m-DCTformer with arbitrary-scale super-resolution approaches for FLOPs, parameters, model capacity, and time.

	FLOPs (G)	**Parameters (M)**	**Model Capacity (MB)**	**Time (ms)**
EDSR [19] + CiaoSR [5]	599	42.83	490	1340
EDSR [19] + LTE [28]	618	39.74	454	483
EDSR [19] + ours	538	47.75	155	391

4.6. Ablation Study

This section presents an experiment that investigates the influence of the proposed modules on SR performance in terms of PSNR and SSIM values. As presented in Tables 5 and 6, when training on the DCT domain when no weight modulation was used, m-DCTformer achieved a PSNR of 35.50 dB on the Set5 and Urban 100 dataset. Table 5 denotes quantitative results with the m-DCTformer on the Set5 dataset for ×2.5. Bold represents the best PSNR and SSIM scores. Moreover, Table 6 denotes quantitative results with the m-DCTformer on the Urban 100 dataset for ×2.5. Bold represents the best PSNR and SSIM scores. Adding a weight modulation to this increases the PSNR to 35.53 dB. This result suggests that modulating the trained weight is effective in restoration when performing arbitrary-scale SR. Next, when learning in the DCT domain is applied, the PSNR increases significantly to 0.42 dB. This result demonstrates that learning in the DCT domain robustly restores the lost high-frequency information. Furthermore, weight modulation increases the PSNR by 0.50 dB in addition to DCT domain learning, suggesting that the m-DCTformer is effective for arbitrary-scale SR by restoring high-frequency information lost during image degradation and modulating the weights to the arbitrary-scale through weight modulation. Finally, we performed an ablation study on d, the parameter in Equation (6). In Table 7, we tested ×3.5 on the Set14 dataset by training when d is 10, 20, and 30. We observed that d is robust to high frequency DCT component restoration at 20. Table 7 denotes quantitative results with the m-DCTformer on the Set 14 dataset for ×3.5. Bold represents the best PSNR and SSIM scores.

Table 5. Quantitative results with the m-DCTformer on the Set5 dataset for ×2.5. Bold represents the best PSNR and SSIM scores.

DCT Domain	Weight Modulation	PSNR	SSIM
		35.50	0.9335
	✓	35.52	0.9335
✓		35.93	0.9424
✓	✓	**35.99**	**0.9426**

Table 6. Quantitative results with the m-DCTformer on the Urban 100 dataset for ×2.5. Bold represents the best PSNR and SSIM scores.

DCT Domain	Weight Modulation	PSNR	SSIM
		29.58	0.8841
	✓	29.61	0.8842
✓		29.84	0.8899
✓	✓	**29.91**	**0.8917**

Table 7. Quantitative results with the m-DCTformer on the Set 14 dataset for ×3.5. Bold represents the best PSNR and SSIM scores.

The Parameter d	Set14	
	PSNR	SSIM
10	29.38	0.8128
20	**29.49**	**0.8134**
30	29.46	0.8133

5. Discussion

In this section, we discuss the limitations of our proposed model. First, m-DCTformer relies on the existing integer-scale SR model. Since we adopt the existing integer-scale SR model as the backbone network, it is highly dependent on the performance of integer-scale SR. In future research, it may be beneficial to study models that directly perform arbitrary-scale SR without using the integer-scale SR model. Second, our model is limited to bicubic interpolation. The LR images used for training and testing were bicubic interpolations. These bicubic LR images may not be able to utilize high-level semantic information because they only perform pixel-level interpolation while ignoring structural features. Therefore, it may be helpful for future research to utilize a variety of low-resolution images instead of being limited to bicubic low-resolution images.

6. Conclusions

This paper proposes an m-DCTformer based on a transformer that learns high-frequency DCT components. The proposed m-DCTformer takes the SR method with an integer scale as the backbone and processes the remaining arbitrary-scale. The depthwise multi-head attention and depthwise feed-forward network proposed by the m-DCTformer are learned when the remaining arbitrary-scale processed by the DCT domain is at a maximum, and they restore the lost high-frequency components in the high-frequency image as input. Each Q, K, and V forms an attention map to restore the feed-forward network. In addition, a weight modulation is learned when the remaining processed residual arbitrary-scale is the minimum to modulate the weights learned at the maximum. The learned weight modulation modulates the weights of Q, K, and V in depthwise multi-head attention. In conclusion, m-DCTformer can manage memory efficiently, and we demonstrated its performance through a flexible combination with the existing integer-scale SR model.

Author Contributions: Conceptualization, S.B.Y.; methodology, M.H.K. and S.B.Y.; software, M.H.K.; validation, M.H.K.; formal analysis, M.H.K. and S.B.Y.; investigation, M.H.K. and S.B.Y.; resources, M.H.K. and S.B.Y.; data curation, M.H.K. and S.B.Y.; writing—original draft preparation, M.H.K. and S.B.Y.; writing—review and editing, M.H.K. and S.B.Y.; visualization, S.B.Y.; supervision, S.B.Y.; project administration, S.B.Y.; funding acquisition, S.B.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Industrial Fundamental Technology Development Program (No. 20018699) funded by MOTIE of Korea and the IITP grant funded by the Korea government (MSIT) (No. 2021-0-02068, RS-2023-00256629).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Zhang, Y.; Huang, Y.; Wang, K.; Qi, G.; Zhu, J. Single image super-resolution reconstruction with preservation of structure and texture details. *Mathematics* **2023**, *11*, 216. [CrossRef]
2. Cha, Z.; Xu, D.; Tang, Y.; Jiang, Z. Meta-Learning for Zero-Shot Remote Sensing Image Super-Resolution. *Mathematics* **2023**, *11*, 1653. [CrossRef]
3. Dong, C.; Loy, C.C.; He, K.; Tang, X. Learning a deep convolutional network for image super-resolution. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; pp. 184–199.
4. Zhang, Y.; Li, K.; Li, K.; Wang, L.; Zhong, B.; Fu, Y. Image super-resolution using very deep residual channel attention networks. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 286–301.
5. Cao, J.; Wang, Q.; Xian, Y.; Li, Y.; Ni, B.; Pi, Z.; Zhang, K.; Zhang, Y.; Timofte, R.; Van Gool, L. Ciaosr: Continuous implicit attention-in-attention network for arbitrary-scale image super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 20–22 June 2023; pp. 1796–1807.
6. Yao, J.E.; Tsao, L.Y.; Lo, Y.C.; Tseng, R.; Chang, C.C.; Lee, C.Y. Local Implicit Normalizing Flow for Arbitrary-Scale Image Super-Resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 20–22 June 2023; pp. 1776–1785.
7. Song, G.; Sun, Q.; Zhang, L.; Su, R.; Shi, J.; He, Y. OPE-SR: Orthogonal Position Encoding for Designing a Parameter-free Upsampling Module in Arbitrary-scale Image Super-Resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 20–22 June 2023; pp. 10009–10020.
8. Yun, J.S.; Yoo, S.B. Single image super-resolution with arbitrary magnification based on high-frequency attention network. *Mathematics* **2021**, *10*, 275. [CrossRef]
9. Ahmed, N.; Natarajan, T.; Rao, K.R. Discrete cosine transform. *IEEE Trans. Comput.* **1974**, *100*, 90–93. [CrossRef]
10. Ghosh, A.; Chellappa, R. Deep feature extraction in the DCT domain. In Proceedings of the 2016 23rd International Conference on Pattern Recognition, Cancun, Mexico, 4–8 December 2016; pp. 3536–3541.
11. Kim, M.H.; Yun, J.S.; Yoo, S.B. Multiregression spatially variant blur kernel estimation based on inter-kernel consistency. *Electron. Lett.* **2023**, *59*, e12805. [CrossRef]
12. Yun, J.S.; Na, Y.; Kim, H.H.; Kim, H.I.; Yoo, S.B. HAZE-Net: High-Frequency Attentive Super-Resolved Gaze Estimation in Low-Resolution Face Images. In Proceedings of the Asian Conference on Computer Vision, Macau SAR, China, 4–8 December 2022; pp. 3361–3378.
13. Yun, J.S.; Yoo, S.B. Kernel-attentive weight modulation memory network for optical blur kernel-aware image super-resolution. *Opt. Lett.* **2023**, *48*, 2740–2743. [CrossRef] [PubMed]
14. Na, Y.; Kim, H.H.; Yoo, S.B. Shared knowledge distillation for robust multi-scale super-resolution networks. *Electron. Lett.* **2022**, *58*, 502–504. [CrossRef]
15. Lee, S.J.; Yoo, S.B. Super-resolved recognition of license plate characters. *Mathematics* **2021**, *9*, 2494. [CrossRef]

16. Dong, C.; Loy, C.C.; He, K.; Tang, X. Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 295–307. [CrossRef] [PubMed]
17. Dong, C.; Loy, C.C.; Tang, X. Accelerating the super-resolution convolutional neural network. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; pp. 391–407.
18. Kim, J.; Lee, J.K.; Lee, K.M. Accurate image super-resolution using very deep convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1646–1654.
19. Lim, B.; Son, S.; Kim, H.; Nah, S.; Mu Lee, K. Enhanced deep residual networks for single image super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 22–25 July 2018; pp. 136–144.
20. Shi, W.; Caballero, J.; Huszár, F.; Totz, J.; Aitken, A.P.; Bishop, R.; Rueckert, D.; Wang, Z. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1874–1883.
21. Zhang, Y.; Tian, Y.; Kong, Y.; Zhong, B.; Fu, Y. Residual dense network for image super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 2472–2481.
22. Dai, T.; Cai, J.; Zhang, Y.; Xia, S.T.; Zhang, L. Second-order attention network for single image super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 11065–11074.
23. Lugmayr, A.; Danelljan, M.; Van Gool, L.; Timofte, R. Srflow: Learning the super-resolution space with normalizing flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020, Proceedings, Part V*; Springer International Publishing: Cham, Switzerland, 2020; pp. 715–732
24. 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, Nashville, TN, USA, 19–21 June 2021; pp. 1833–1844.
25. Hu, X.; Mu, H.; Zhang, X.; Wang, Z.; Tan, T.; Sun, J. Meta-SR: A magnification-arbitrary network for super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 1575–1584.
26. Son, S.; Lee, K.M. SRWarp: Generalized image super-resolution under arbitrary transformation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–21 June 2021; pp. 7782–7791.
27. Wang, L.; Wang, Y.; Lin, Z.; Yang, J.; An, W.; Guo, Y. Learning a single network for scale-arbitrary super-resolution. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Nashville, TN, USA, 19–21 June 2021; pp. 4801–4810.
28. Lee, J.; Jin, K.H. Local texture estimator for implicit representation function. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 21–24 June 2022; pp. 1929–1938.
29. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–15.
30. Hendrycks, D.; Gimpel, K. Gaussian error linear units (gelus). *arXiv* **2016**, arXiv:1606.08415.
31. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
32. Chen, X.; Wang, X.; Zhou, J.; Dong, C. Activating more pixels in image super-resolution transformer. *arXiv* **2022**, arXiv:2205.04437.
33. 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, 22–25 July 2017; pp. 126–135.
34. Bevilacqua, M.; Roumy, A.; Guillemot, C.; Alberi-Morel, M.L. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In Proceedings of the 23rd British Machine Vision Conference, Surrey, UK, 3–7 September 2012.
35. Zeyde, R.; Elad, M.; Protter, M. On single image scale-up using sparse-representations. In Proceedings of the Curves and Surfaces: 7th International Conference, Avignon, France, 24–30 June 2010; pp. 711–730.
36. 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, 8–10 June 2015; pp. 5197–5206.
37. Lugmayr, A.; Danelljan, M.; Timofte, R. Ntire 2020 challenge on real-world image super-resolution: Methods and results. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 16–18 June 2020; pp. 494–495.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.