Dynamic High-Pass Filtering and Multi-Spectral Attention for Image Super-Resolution

Salma Abdel Magid[1], Yulun Zhang[2], Donglai Wei[3], Won-Dong Jang[1],
Zudi Lin[1], Yun Fu[2], Hanspeter Pfister[1]
[1]Harvard University [2]Northeastern University [3]Boston College

Abstract

Deep convolutional neural networks (CNNs) have pushed forward the frontier of super-resolution (SR) research. However, current CNN models exhibit a major flaw: they are biased towards learning low-frequency signals. This bias becomes more problematic for the image SR task which targets reconstructing all fine details and image textures. To tackle this challenge, we propose to improve the learning of high-frequency features both locally and globally and introduce two novel architectural units to existing SR models. Specifically, we propose a dynamic high-pass filtering (HPF) module that locally applies adaptive filter weights for each spatial location and channel group to preserve high-frequency signals. We also propose a matrix multi-spectral channel attention (MMCA) module that predicts the attention map of features decomposed in the frequency domain. This module operates in a global context to adaptively recalibrate feature responses at different frequencies. Extensive qualitative and quantitative results demonstrate that our proposed modules achieve better accuracy and visual improvements against state-of-the-art methods on several benchmark datasets.

1. Introduction

Image SR is a modeling task that estimates a high-resolution (HR) image from its low-resolution (LR) counterpart. Image SR is a challenging and ill-posed problem since multiple solutions exist for any LR input. Given the recent advances in deep learning, convolutinal neural network (CNN) based SR methods have been leveraged in a wide variety of research domains such as biomedicine, object recognition, and hyper-spectral imaging [9, 21, 32, 43].

The promising results and potential impact of SR in these domains have garnered attention from the vision research community. Many CNN-based methods have been proposed [4, 5, 6, 7, 17, 20, 47, 49] and significantly outperform traditional methods. In line with the 'very deep' paradigm, these methods use over-parameterized networks with hundreds of layers. This approach is usually coupled with a recent architectural breakthrough known as residual learning. Residual learning alleviates the degradation problem due to increased depth, and simplifies the learning task, which improves network convergence.

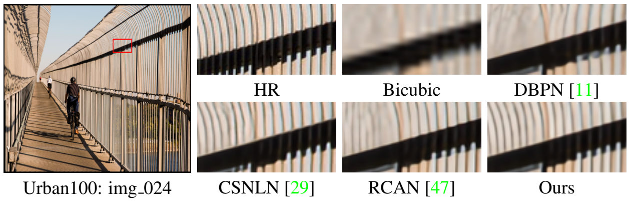

Figure 1: Visual comparison (×4) on "image_024" from Urban100. Existing methods suffer from blurring artifacts.

Although these advancements have enhanced performance and are now commonplace in SR networks, these methods still suffer from a serious flaw (see Figure 1). It has been demonstrated that neural networks exhibit a bias towards low-frequency signals. Figure 2 illustrates a prime example of this. In the output of a popular and robust SR baseline, RCAN [47], we can see that the high-frequency data are significantly reduced, causing the reconstruction to be overly smooth. This is due to many aspects of training, such as the loss function, architecture type, and optimization method. Ledig *et al*. [20] already showed that standard pixel-wise metrics (ℓ_1 or ℓ_2) tend to pull the reconstruction towards an average of the possible reconstructions equidistant in terms of the ℓ_2 loss on the natural image manifold.

Similarly, higher frequencies struggle to propagate due to the architecture and optimization method of the networks [2]. They become quickly saturated with low-frequency patterns first, thereby halting the learning of additional information. Since there is high information redundancy between channels, many recent works propose using various attention mechanisms to re-weight channels. The classic channel attention mechanism, SENet [13], suffers from one major drawback. Qin *et al*. [33] theoretically demonstrated that by using global average pooling, SENet discards all other frequencies except the lowest one. Another issue

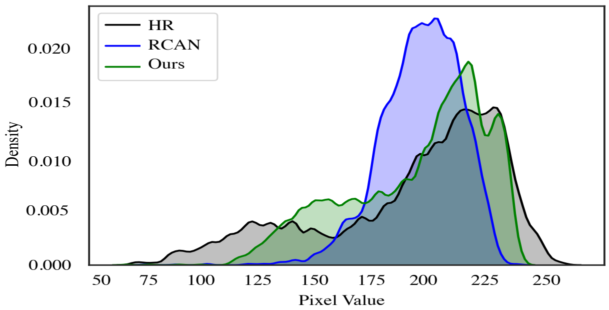

Figure 2: Comparison of distributions for a sample of sequential pixels sampled from the patches shown in Figure 1. Existing methods produce an overly smooth distribution.

that arises is aliasing, the phenomenon that high-frequency signals degenerate after sampling. This is due to downsampling layers, which are widely used in deep networks to reduce parameters and computation [51]. When we consider image SR applications, these flaws are exacerbated since the modeling task requires high-frequency information to be complete.

Motivated by these issues, we propose to bridge this divide by ensuring that high-frequency information propagates through the network. We tackle this problem both locally and globally. Our global approach is to modify the existing channel attention mechanism by utilizing a broader frequency spectrum in contrast to existing methods. This increases the representational power of the network and preserves the inter-dependencies between features. We propose to amplify high-frequency details in a dynamic and context-aware fashion in addition to a novel channel attention mechanism. We learn a different high-pass filtering kernel for each spatial location, which is then applied to the input features at their respective location. Low-frequency information is preserved via long- and short-range skip connections. By following the convolution operation with a high-pass filtering operation, we pivot the network's learning capacity to more difficult high-frequency details.

In summary, our main contributions are as follows:

- We propose a dynamic high-pass filtering layer for image super-resolution (SR) networks. This module enhances the network's discriminative learning ability by enabling it to focus on useful spatial content.
- We further propose a matrix multi-spectral channel attention mechanism that predicts the attention map of features decomposed in the frequency domain. The feature channels are then adaptively rescaled based on their maximum frequency response.
- We provide visual results and analyses regarding our proposed modules. We also conduct extensive comparisons with recent image SR methods and achieve significant gains quantitatively and visually.

2. Related Work

Image Super-resolution. State-of-the-art deep learning-based SR methods postulate the problem as a dense regression task that learns an end-to-end mapping represented by a deep CNN between low-resolution and high-resolution images. The pioneering work by Dong *et al*. [6] first utilized deep learning to solve the SR problem using a three-layer CNN and further improved the training efficiency in follow-up work [7]. Following this first attempt, many works have achieved better performance by using the "very deep paradigm" that increases the depth and width of the CNNs and integrates residual learning [17, 24, 47, 49]. More recent works integrate different channel and spatial attention mechanisms to utilize the interaction of different layers, channels, and positions. Dai *et al*. [4] propose SAN, which includes an attention module to learn feature interdependencies by considering second-order statistics of features. Niu *et al*. proposed HAN [30], which includes both a layer attention module and a channel-spatial attention module, to emphasize hierarchical features by considering the correlations among layers. RBAN [5] consists of two types of attention modules for feature expression and feature correlation learning. We differ from these works by explicitly focusing on the learning of high-frequency signals.

Visual Attention. SENet [13] accomplishes channel attention using a single global descriptor for each channel by globally average pooling. These descriptors are then passed to a multi-layer perceptron (MLP) to calculate the weights of each channel. Several works have extended this original scheme by also integrating spatial attention, including CBAM [41], DAN [10] and scSE [34]. Additional works incorporate a variety of techniques to reduce redundancy of the fully connected layers in the MLP (ECANet) [39] and to selectively aggregate channels (SKNet) [22].

However, most of these methods use only the lowest frequency component (via averaging) of the features' frequency spectrum, as theoretically demonstrated in Qin *et al*. [33]. To overcome this, FCA [33] builds on the original SENet by proposing a frequency-based approach to channel attention. This is done by grouping channels and assigning the same single frequency to each channel in a given group. The global descriptor for each channel is its corresponding frequency coefficient calculated via the Discrete Cosine Transform. In this way, they expand the frequencies being utilized by the attention mechanism. We adapt and improve this mechanism to image SR by considering multiple frequency components for each channel.

Adaptive Filtering Layer. Image filtering is a classic computer vision technique in image restoration tasks, including super-resolution, de-noising, and in-painting [36]. Previous works have integrated classic filters (*e.g*., Gaussian) into deep models to tackle vision tasks at different lev-

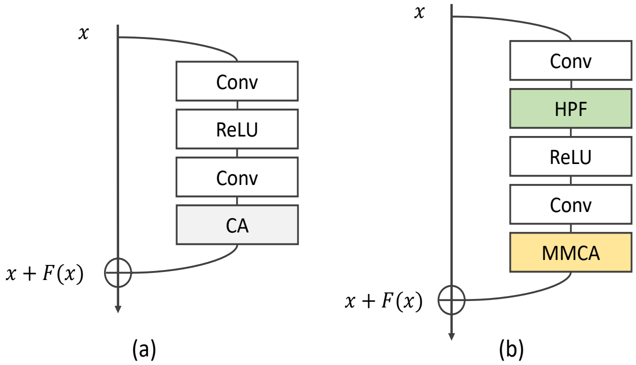

Figure 3: Comparison of residual blocks in original (a) RCAN [47] and (b) ours. We add our dynamic high-pass filtering (HPF) layer after the first convolution and replace the standard channel attention with our modified multi-spectral channel attention (MMCA).

els [14, 42, 46]. However, those filters have fixed elements, restricting the adaptation to specific spatial locations and image content. Moreover, these filters require careful tuning of hyper-parameters. Therefore, recent works also make the filters learnable during optimization and spatially-varying based on local features [16, 35, 51]. Specifically, Zou *et al*. [51] restrict the learned filters to be *low-pass* to counter the aliasing artifacts in model downsampling layers. We incorporate the adaptive filtering idea into super-resolution models and introduce the dynamic *high-pass* filtering (HPF) layer. The novel HPF layer can better preserve the high-frequency signals in deep models, which is favorable for the SR task since it requires fine details and textures.

3. Proposed Method

In this section, we introduce two novel modules that can be seamlessly integrated into existing SR architectures (*e.g*., RCAN [47]) to improve the performance in super-resolution, including the *Matrix Multi-Spectral Channel Attention* (MMCA) module (Sec. 3.2) and the *Dynamic High-Pass Filtering Layer* (HPF) module (Sec. 3.1). These modules conduct local and global frequency modulation dynamically. HPF amplifies the high-frequencies of input features by dynamically learning and applying different high-pass kernels for each spatial location. MMCA then relatively rescales channels using their maximal frequency response. Figure 3 demonstrates how these modules are integrated into a standard residual block used in image SR networks.

3.1. Dynamic High-Pass Filtering Layer (HPF)

In the high-pass filtering (HPF) layer we learn to dynamically generate different spatial and channel high-pass kernels which are then applied to their respective locations. Using the same kernel across the spatial extent of the input features may not accurately capture all the high frequency details since the frequency of a signal can vary dramatically across spatial locations. Consequently, we learn a different high-pass kernel for each spatial location. In a similar vein, we can also learn a different kernel for each channel. This would incur severe computational overhead. It is sufficient to split the channels into groups since there is information redundancy in channel features. Thus, we split the C channels into g groups and predict a different set of high-pass kernels for each group. Figure 4 illustrates our HPF module. Given an input $X \in R^{H \times W \times C}$, we learn g kernels for each spatial location (i, j) of X then apply these kernels to X in their respective local locations and groups to produce our output Y. Note that for each spatial location (i, j) there are a set of points (indicated by gray boxes overlaid on X in Figure 4) surrounding it which are involved in the application of kernel w_g. This technique enables us to propagate high-frequencies to the subsequent layer. By using this module throughout the depth of the network, we can preserve the high-frequency information.

To learn the filters, we follow [51] by applying a standard convolution followed by a batch normalization to the input feature X, where $X \in R^{N \times H \times W \times C}$. This produces our kernels, w where $w \in R^{n \times g \times k^2 \times h \times w}$. In [51], the authors ensure their filters are low-pass by constraining the weights to be positive and sum to one by applying the softmax function. To produce the corresponding high-pass kernel, we simply invert this by subtracting the low-pass kernel from the identity kernel.

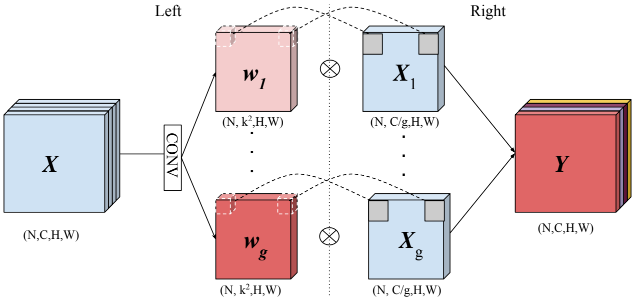

Figure 4: Our proposed dynamic HPF module. For each group of channels, we predict a different $k \times k$ high pass kernel for each spatial location. The kernels are then applied to their respective locations to produce the final output.

3.2. Matrix Multi-Spectral Channel Attention

Channel Attention (CA). After amplifying high frequency details in the feature extraction layers of the residual block, we next operate in a global context by using CA. Recall that the standard approach, SENet [13], calculates the average of each channel using global average pooling (GAP). We revisit theoretical findings [33] and demonstrate that this approach is only using the lowest frequency information of the input features. Thus, any image enhancement(*i.e*., super

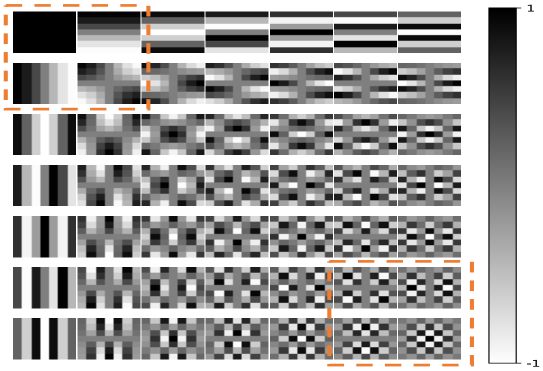

Figure 5: Visualization of the DCT basis functions. Orange boxes (top-left and bottom-right) indicate the chosen frequency components for the MMCA module.

resolution, de-blurring, de-noising, etc.) network that uses CA is discarding other potentially useful high frequency information for image reconstruction. We claim that these high frequency components carry valuable information. As such, we propose a modified CA mechanism that uses several frequency components for each channel.

Transformation to Frequency Domain. There are several transformation methods one can use to decompose a signal to its spatial frequency spectrum. The predominant method for frequency analysis is the Discrete Fourier Transform (DFT). Although this is widely used, we will instead focus on another attractive method due to its simplicity, the Discrete Cosine Transform (DCT) [1]. The DCT uses a sum of cosine functions oscillating at different frequencies to express a set of data points. One can view the DCT as a special case of the DFT by only considering the real components of the decomposition. The DCT has a unique property which makes it the heart of the most widely used image compression standard and digital image format. The DCT has strong “energy compaction” which implies that a large proportion of the total signal energy is contained in a handful of coefficients. This is especially true for natural image data where there is generally large regions of uniform signal.

For an input $x \in \mathbb{R}^{H \times W}$ where H is the height of x, and W is the width of x, the 2D DCT frequency spectrum, $g \in \mathbb{R}^{H \times W}$ is defined as:

$$g_{h,w} = \sum_{p=0}^{H-1} \sum_{q=0}^{W-1} x_{p,q} \underbrace{\cos(\frac{\pi h}{H}(p + \frac{1}{2})) \cos(\frac{\pi w}{W}(q + \frac{1}{2}))}_{\text{DCT weights}},$$

$$s.t. \;\; h \in \{0, 1, \cdots, H-1\}, w \in \{0, 1, \cdots, W-1\}, \tag{1}$$

For simplicity, we omit normalizing constants which do not affect the results. As discussed in FCANet [33] this decomposition produces coefficients $g_{h,w}$ which are simply a weighted sum of the input. The parameters h and w control the frequency of the cosine functions. Suppose h and w in Eq. 1 are 0, we have:

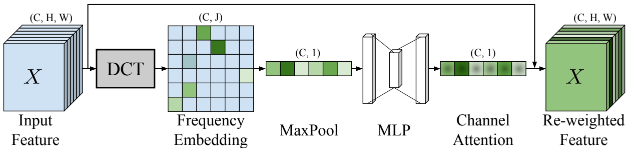

Figure 6: Our MMCA module. The input feature is first transformed to the frequency domain using the discrete cosine transform. The resulting matrix is max-pooled then fed as input to an MLP which provides the channel attention.

$$\begin{aligned} g_{0,0} &= \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} x_{i,j} \cos(\frac{0}{H}(i + \frac{1}{2})) \cos(\frac{0}{W}(j + \frac{1}{2})) \\ &= \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} x_{i,j} \\ &= \mathrm{GAP}(x) \cdot H \cdot W. \end{aligned} \tag{2}$$

If we set $h = 0$, $w = 0$, then we can see that the cosine terms evaluate to 1, and we are simply summing the input (and dividing by a normalizing factor). In Eq. 2, $g_{0,0}$ represents the lowest frequency component of the 2D DCT, and it is proportional to GAP.

Matrix Multi-Spectral CA. We approach the design of our CA mechanism using these findings. Since our goal is to utilize more of the frequency spectrum of the features, we transform our input to a frequency embedding using the DCT. The global descriptor for each channel is then the maximum frequency response. We provide additional technical details below.

The benefit of using the DCT is that we can pre-compute the DCT weights as a pre-processing step. That way, during training and testing, there is little additional overhead. The specifics of our method are described in Figure 6. Suppose for each channel, C, in our input features X, where $X \in \mathbb{R}^{C \times H \times W}$ we want to use J frequency components. We pre-compute the matrix of DCT weights, $\mathrm{A} \in \mathbb{R}^{J \times C \times H \times W}$ using equation (1). That is, for the r^{th} frequency component g_{uv}, we calculate $A_{r,:,i,j} = \cos\left(\frac{\pi u(2i+1)}{2H}\right) \cos\left(\frac{\pi v(2j+1)}{2W}\right)$ where $r \in 0, 1, 2, ..J$. Note that r corresponds to a specific component (u, v),

Expanding X such that $X \in \mathbb{R}^{1 \times C \times H \times W}$ then performing element wise multiplication followed by a spatial sum produces our DCT coefficients. These coefficients are our J global descriptors. More specifically: $D = \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} X_{:,:,i,j} \odot A^{T}_{:,:,i,j}$ where $D \in \mathbb{R}^{C \times J}$.

To reduce the matrix of frequency global descriptors, we take the maximum frequency response for each channel, C. We then apply the function $F(x)$ where F corresponds to

Table 1: Ablation study on the number of frequency components. Evaluated using Urban100 at ×4 scale.

#	1	2	4	8	16
PSNR	26.24	26.36	26.38	**26.39**	26.33

FC layers followed by a standard sigmoid denoted by the function $S(x)$, where $S(x) = \frac{1}{1+e^{-x}}$ as follows:

$$attn_c = S(F(\max_j D_c)).$$

Finally, the input features X are re-weighted using the final calculated attention. Thus, each of the J frequencies contributes to the final attention.

4. Experiments

4.1. Settings

Datasets. There are a variety of datasets for image SR with varying image content, resolution, and quality. To train and test our model, we use the DIV2K [38] image dataset. DIV2K is a newly proposed, rich image dataset consisting of 800 training images, 100 testing images, and 100 validation images. To enrich the training set with more diverse textures, we also use the Flickr2K dataset [24]. For testing, we use five standard benchmark datasets: Set5 [3], Set14 [44], B100 [27], Urban100 [15], and Manga109 [28].

Evaluation Metrics. To evaluate our method, we follow standard practice and report the peak signal-to-noise-ratio (PSNR) and the structural similarity metric (SSIM) [40]. These metrics are applied to the Y channel (*i.e.* luminance) of the the transformed RGB images in the YCbCr space.

Training Settings. To train our models, a batch of 16 LR RGB images are randomly sampled and cropped to a size of 48×48. Training patches are augmented using random horizontal flips and 90° rotation. Our models are trained using the ADAM optimizer [18] by setting β_1=0.9, β_2=0.99 and ϵ=10^{-8}. The initial learning rate is set to 10^{-4} and halved every 200 epochs. We use the ℓ_1 loss since it has been empirically demonstrated to outperform the ℓ_2 loss for image SR tasks. The model is implemented in PyTorch [31] and trained using a single Nvidia V100 GPU.

We integrate our proposed modules, HPF and MMCA, into RCAN [47]. RCAN consists of 10 residual groups (RG) which each contain 20 residual blocks (RB). The number of channels is set to 64. To reduce the computational overhead, we place our components only in the last RB of each RG of RCAN. The HPF module is added after the first convolution as illustrated in Figure 3 while the CA is swapped with our MMCA. We set the number of groups in the HPF to 8. The number of frequency components for each channel is also 8. The chosen components are a combination of high and low frequencies. These hyper-parameter settings are discussed in more detail in their corresponding ablation study subsections below. To compute the frequency coefficients, we first adaptively down-sample the input channels to a spatial extent of 7×7 similar to [33].

Table 2: Ablation study on the number of HPF modules in a standard residual block. Evaluated using Manga109 benchmark at ×4 scale.

#	0	1	2
PSNR	30.65	**30.82**	30.74

Table 3: Ablation study on the number of groups in the HPF module. Evaluated using Manga109 at ×4 scale.

#	2	4	8	16
PSNR	30.82	30.79	30.82	**30.88**

4.2. Ablation Studies

Position of HPF in the Standard Residual Block. To determine where and how many HPF layers to place in the standard residual block (RB), we conducted an ablation study. Figure 6 illustrates the positioning of the HPF layer within a RB as following the first convolution layer. Alternatively, we could create symmetrical operation by placing another HPF layer after the second convolution as well such that each convolution is followed by a high-pass filtering operation. However, our experiments in table 2 demonstrate that adding a single HPF is sufficient. This also shows the effectiveness of the layer is not simply due to increasing the number of parameters.

Number of HPF Groups. To study the influence of the number of groups in the HPF module, we conduct an ablation study by varying the groups hyper-parameter. Table 3 demonstrates that increasing the number of groups generally leads to improved performance. Since we compute a different set of filters for each group, this computation can be expensive as the depth of the network increases (i.e. more residual blocks). To alleviate this, we take a middle ground by using 8 groups since there is little performance difference and is computationally more efficient. In this way, the learned filters can adapt to different frequencies across feature channels, while saving computational costs by learning the same filter per group.

HPF Filter Analysis. To better understand the behavior of the HPF module, we analyze the learned filters. What differentiates various filters is their variance. For example, a $k \times k$ smoothing filter, also known as the average filter, has a variance of zero since it consists of equivalent elements each with a value of $\frac{1}{k}$. A smaller variance corresponds to filtering lower frequencies whereas a larger variance corresponds to filtering higher frequencies.

Figure 7 visualizes variance of the learned filter weights across different spatial locations. HPF module learns filters that spatially adapt to different image content. For example, in the first image of the bird in figure 7, there is high variance precisely where there are abrupt and sharp transitions at the leaf edges. Similarly, in the image of the building, there are several edges and pixel intensity fluctuations which our HPF filters are able to amplify. Thus, the learned filters can propagate high frequency details after the convolution while preserving useful image content. We can also see that the filters are able to capture higher frequency information with sharp intensity transitions while attenuating the lower frequency details such as the uniform background.

Number of Frequency Components. Table 1 compares the effect of using multiple frequency components in the channel attention module. The general trend is clear: increasing the number of frequency components will increase performance. However, at a certain point (16 frequency components in Table 1) the performance stagnates. All experiments using more than a single frequency component in our modified frequency based channel attention show a large performance gap when compared to the standard channel attention. We claim that this is due to the fact that only using one frequency components discards useful information. The additional features encode other salient information and can compensate the “soft” global statistics encoded by average-pooling. Consequently, pooling features based on their frequency results in meaningful global descriptors. This verifies our claim that adding additional frequency information aids the network in integrating more components from the wider frequency spectrum. Given these results, we use 8 components for our final models.

The chosen frequency components are illustrated in Figure 5. Moving across the rows and columns of the DCT grid of basis functions in figure 5 corresponds to oscillating more either in the vertical or horizontal directions. Intuitively, the top left corner corresponds to zero oscillations in either directions (i.e., $h = 0, w = 0$ in equation (1)) which results in a constant term. On the other hand, the highest vertical and horizontal frequency component is in the bottom right corner. Choosing components in these corners of the DCT matrix, we provide a diverse spectrum for MMCA module.

Comparison with Other Attention Mechanisms. We compare our method with the standard SENet and FCA. As demonstrated in table 4, our modified frequency channel attention outperforms both baselines. By incorporating a wider frequency spectrum of the input features, we are able to adaptively re-weight the channels which in turn enables a performance boost. The key difference between our method and FCA is that FCA groups channels and assigns the same frequency to each channel in a group. On the other hand, our method instead computes J frequency coefficients for each channel and selects the maximum frequency response for each channel. In this way, we are able to capture and focus on the high frequencies. Additionally, we can view the choice of frequencies as a toggle by which we expand the spectrum. By replacing standard CA and FCA with our MMCA, we can better leverage higher frequencies.

Table 4: Comparison with other attention mechanisms in image SR. Evaluated by PSNR using Urban100 and Manga109 benchmarks at ×4 scale.

Module	SENet	FCA	MMCA (Ours)
Urban100	26.24	26.29	**26.39**
Manga109	30.65	30.67	**30.77**

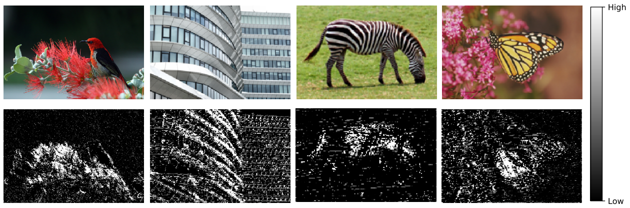

Figure 7: Variance of learned dynamic high-pass kernel from the 4^{th} residual block, 5^{th} group. The kernel correctly learns to filter high-frequency details such as sharp pixel value transitions.

4.3. Comparison with State-of-the-Art Methods

We extensively compare our method with 17 state-of-the-art image SR methods in table 5. For qualitative comparisons, we compare with 7 state-of-the-art methods in very challenging cases.

Quantitative Results. Table 5 shows quantitative comparisons for ×2, ×3, and ×4 results. As demonstrated in Table 5, our model outperforms the compared methods across scales and benchmarks. The consistently higher PSNR and SSIM values provide promising potential to investigating the frequency domain for image SR. Our method reaches a maximum PSNR increase of 0.52 dB for the ×2 scale, 0.48 dB for the ×3 scale, and 0.45 dB for the ×4 scale. The maximum PSNR increase indicates the maximum difference between our method and the second-best method that occurs over all datasets for a given scale.

As previously mentioned, we use RCAN as our SR backbone. Consequently, when we compare the number of parameters between our modified model and RCAN, they are roughly equivalent. Although this is the case, our model outperforms RCAN by maximum PSNR increases of 0.54 dB for the ×2 scale , 0.63 dB for the ×3 scale, and 0.66 dB for the ×4 scale. By modifying the last RB of each RG in RCAN to that of Figure 3 (b), we are able to focus on more informative features and amplify high frequency details. This observation indicates that the HPF and MMCA mod-

Table 5: Quantitative comparison with other state-of-the-art methods. Average PSNR (dB) and SSIM for scale factor ×2, ×3 and ×4 are shown for several benchmarks. Best and second best performance are **bolded** and underlined, respectively.

Method	Scale	Set5		Set14		B100		Urban100		Manga109	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
LapSRN [19]	×2	37.52	0.9591	33.08	0.9130	31.08	0.8950	30.41	0.9101	37.27	0.9740
MemNet [37]	×2	37.78	0.9597	33.28	0.9142	32.08	0.8978	31.31	0.9195	37.72	0.9740
EDSR [24]	×2	38.11	0.9602	33.92	0.9195	32.32	0.9013	32.93	0.9351	39.10	0.9773
SRMDNF [45]	×2	37.79	0.9601	33.32	0.9159	32.05	0.8985	31.33	0.9204	38.07	0.9761
DBPN [11]	×2	38.09	0.9600	33.85	0.9190	32.27	0.9000	32.55	0.9324	38.89	0.9775
RDN [49]	×2	38.24	0.9614	34.01	0.9212	32.34	0.9017	32.89	0.9353	39.18	0.9780
RCAN [47]	×2	38.27	0.9614	34.12	0.9216	32.41	0.9027	33.34	0.9384	39.44	0.9786
NLRN [25]	×2	38.00	0.9603	33.46	0.9159	32.19	0.8992	31.81	0.9249	N/A	N/A
RNAN [48]	×2	38.17	0.9611	33.87	0.9207	32.31	0.9014	32.73	0.9340	39.23	0.9785
SRFBN [23]	×2	38.11	0.9609	33.82	0.9196	32.29	0.9010	32.62	0.9328	39.08	0.9779
OISR [12]	×2	38.21	0.9612	33.94	0.9206	32.36	0.9019	33.03	0.9365	N/A	N/A
SAN [4]	×2	38.31	0.9620	34.07	0.9213	32.42	0.9028	33.10	0.9370	39.32	0.9792
CSNLN [29]	×2	38.28	0.9616	34.12	0.9223	32.40	0.9024	33.25	0.9386	39.37	0.9785
RFANet [26]	×2	38.26	0.9615	34.16	0.9220	32.41	0.9026	33.33	0.9389	39.44	0.9783
HAN [30]	×2	38.27	0.9614	34.16	0.9217	32.41	0.9027	33.35	0.9385	39.46	0.9785
NSR [8]	×2	38.23	0.9614	33.94	0.9203	32.34	0.9020	33.02	0.9367	39.31	0.9782
IGNN [50]	×2	38.24	0.9613	34.07	0.9217	32.41	0.9025	33.23	0.9383	39.35	0.9786
Ours	×2	**38.38**	**0.9620**	**34.33**	**0.9232**	**32.50**	**0.9036**	**33.66**	**0.9412**	**39.98**	**0.9798**
LapSRN [19]	×3	33.82	0.9227	29.87	0.8320	28.82	0.7980	27.07	0.8280	32.21	0.9350
MemNet [37]	×3	34.09	0.9248	30.00	0.8350	28.96	0.8001	27.56	0.8376	32.51	0.9369
EDSR [24]	×3	34.65	0.9280	30.52	0.8462	29.25	0.8093	28.80	0.8653	34.17	0.9476
SRMDNF [45]	×3	34.12	0.9254	30.04	0.8382	28.97	0.8025	27.57	0.8398	33.00	0.9403
RDN [49]	×3	34.71	0.9296	30.57	0.8468	29.26	0.8093	28.80	0.8653	34.13	0.9484
RCAN [47]	×3	34.74	0.9299	30.65	0.8482	29.32	0.8111	29.09	0.8702	34.44	0.9499
NLRN [25]	×3	34.27	0.9266	30.16	0.8374	29.06	0.8026	27.93	0.8453	N/A	N/A
RNAN [48]	×3	34.66	0.9290	30.53	0.8463	29.26	0.8090	28.75	0.8646	34.25	0.9483
SRFBN [23]	×3	34.70	0.9292	30.51	0.8461	29.24	0.8084	28.73	0.8641	34.18	0.9481
OISR [12]	×3	34.72	0.9297	30.57	0.8470	29.29	0.8103	28.95	0.8680	N/A	N/A
SAN [4]	×3	34.75	0.9300	30.59	0.8476	29.33	0.8112	28.93	0.8671	34.30	0.9494
CSNLN [29]	×3	34.74	0.9300	30.66	0.8482	29.33	0.8105	29.13	0.8712	34.45	0.9502
RFANet [26]	×3	34.79	0.9300	30.67	0.8487	29.34	0.8115	29.15	0.8720	34.59	0.9506
HAN [30]	×3	34.75	0.9299	30.67	0.8483	29.32	0.8110	29.10	0.8705	34.48	0.9500
NSR [8]	×3	34.62	0.9289	30.57	0.8475	29.26	0.8100	28.83	0.8663	34.27	0.9484
IGNN [50]	×3	34.72	0.9298	30.66	0.8484	29.31	0.8105	29.03	0.8696	34.39	0.9496
Ours	×3	**34.92**	**0.9312**	**30.83**	**0.8507**	**29.42**	**0.8128**	**29.44**	**0.8761**	**35.07**	**0.9525**
LapSRN [19]	×4	31.54	0.8850	28.19	0.7720	27.32	0.7270	25.21	0.7560	29.09	0.8900
MemNet [37]	×4	31.74	0.8893	28.26	0.7723	27.40	0.7281	25.50	0.7630	29.42	0.8942
EDSR [24]	×4	32.46	0.8968	28.80	0.7876	27.71	0.7420	26.64	0.8033	31.02	0.9148
SRMDNF [45]	×4	31.96	0.8925	28.35	0.7787	27.49	0.7337	25.68	0.7731	30.09	0.9024
DBPN [11]	×4	32.47	0.8980	28.82	0.7860	27.72	0.7400	26.38	0.7946	30.91	0.9137
RDN [49]	×4	32.47	0.8990	28.81	0.7871	27.72	0.7419	26.61	0.8028	31.00	0.9151
RCAN [47]	×4	32.63	0.9002	28.87	0.7889	27.77	0.7436	26.82	0.8087	31.22	0.9173
NLRN [25]	×4	31.92	0.8916	28.36	0.7745	27.48	0.7306	25.79	0.7729	N/A	N/A
RNAN [48]	×4	32.43	0.8977	28.83	0.7871	27.72	0.7410	26.61	0.8023	31.09	0.9149
SRFBN [23]	×4	32.47	0.8983	28.81	0.7868	27.72	0.7409	26.60	0.8015	31.15	0.9160
OISR [12]	×4	32.53	0.8992	28.86	0.7878	27.75	0.7428	26.79	0.8068	N/A	N/A
SAN [4]	×4	32.64	0.9003	28.92	0.7888	27.78	0.7436	26.79	0.8068	31.18	0.9169
CSNLN [29]	×4	32.68	0.9004	28.95	0.7888	27.80	0.7439	**27.22**	**0.8168**	31.43	0.9201
RFANet [26]	×4	32.66	0.9004	28.88	0.7894	27.79	0.7442	26.92	0.8112	31.41	0.9187
HAN [30]	×4	32.64	0.9002	28.90	0.7890	27.80	0.7442	26.85	0.8094	31.42	0.9177
NSR [8]	×4	32.55	0.8987	28.79	0.7876	27.72	0.7414	26.61	0.8025	31.10	0.9145
IGNN [50]	×4	32.57	0.8998	28.85	0.7891	27.77	0.7434	26.84	0.8090	31.28	0.9182
Ours	×4	**32.79**	**0.9019**	**29.06**	**0.7922**	**27.87**	**0.7458**	27.17	0.8163	**31.88**	**0.9266**

ules significantly improve the performance. In our model, the last RB of each RG serves as a gate which (1) passes through high frequency details and (2) operates on a broader frequency spectrum when rescaling the outgoing features. Since our modules are operating within a residual group, the low frequency details are preserved via skip connections.

Qualitative Results. In Figure 8, we visually illustrate the qualitative comparisons for several images from the Urban100 benchmark on the ×4 scale. Our model reconstructs images more accurately and clearly than other methods.

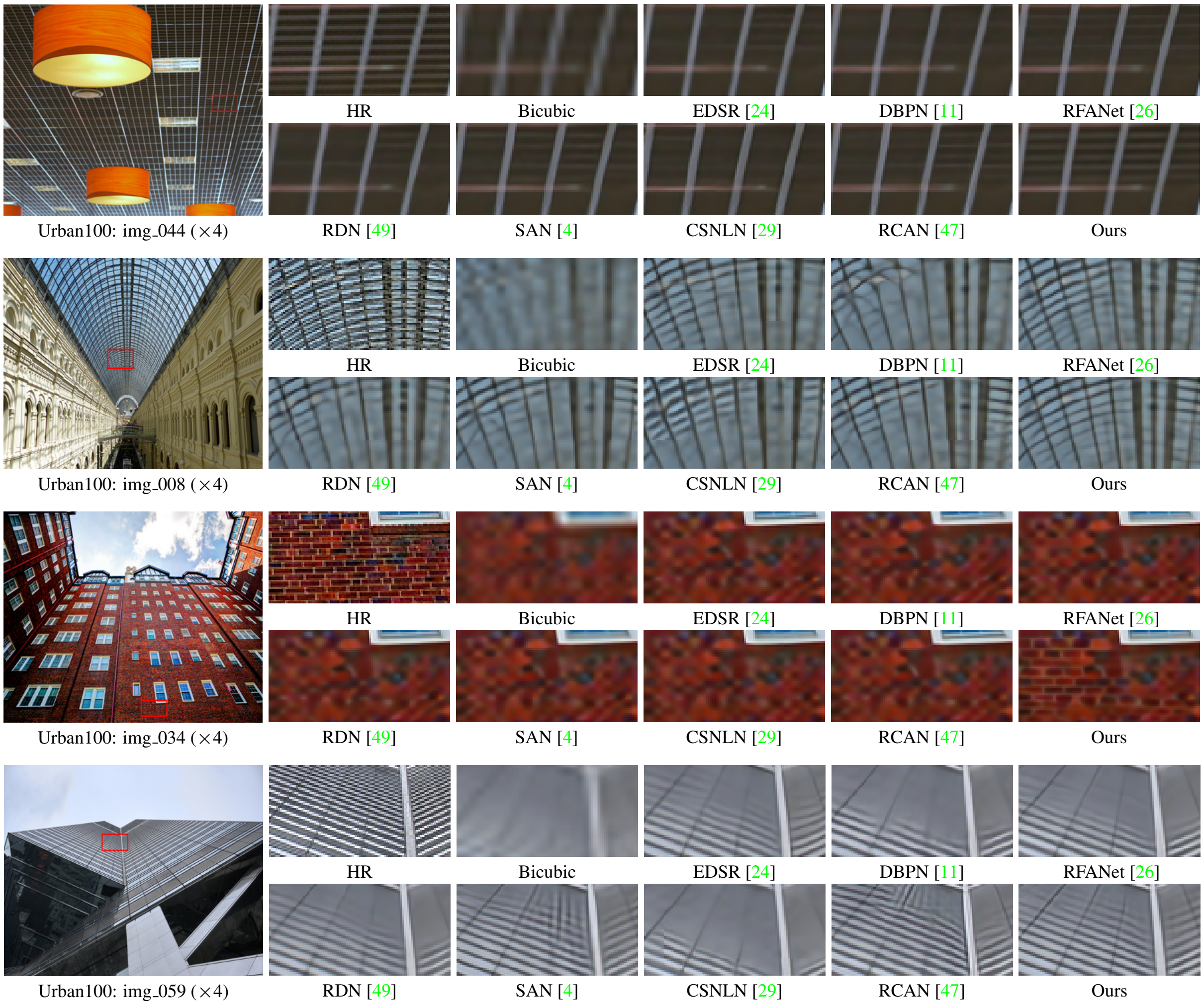

Figure 8: Visual comparison for ×4 SR on Urban100 dataset. Most compared methods suffer from blurring artifacts. Our method is able to reconstruct high-frequency details better than existing methods.

Different patterns and details are correctly produced by our method, while the output of other methods contain blurry patches or artifacts. For example, our method is particularly well-suited for line reconstruction. We can see in "img034" of the Urban100 dataset that our method can correctly produce a subset of the bricks on the wall of the building.

Similarly, in "img059", the horizontal lines are correctly and clearly produced by our method whereas RCAN and SAN produce random vertical stripes which are not present in the ground truth. The remaining methods all suffer from a blurring artifact in this patch.

However, our method is capable of alleviating the blurring artifacts and recovering more high frequency details. Even more so, our method can distinctly delineate several structures as illustrated in "img008" while other methods combine and blur lines in the vertical and/or horizontal direction. These comparisons serve to demonstrate that our modified residual block can extract more sophisticated features from the LR space.

5. Conclusion

This paper introduces the matrix multi-spectral channel attention (MMCA) and dynamic high-pass filtering (HPF) modules to improve the learning of high-frequency features in the image SR task. With the novel and seamless integration of the proposed modules into a standard SR backbone (RCAN), we can sufficiently focus on high-frequency details in input features. Our experiments suggest that following the convolution layer with our dynamic high-pass filtering operation enables preserving essential details and textures. We combine this module with our MMCA to package a new powerful residual block that can be seamlessly integrated into different architectures. There remain some areas for further investigation. For the MMCA module, we need to determine how to appropriately select frequency components. A promising path for further exploration would be to potentially incorporate this in the learning task.

Acknowledgements. This work is partially supported by NIH award 5U54CA225088-03 and by NSF award IIS-1835231.

References

[1] Nasir Ahmed, T₋ Natarajan, and Kamisetty R Rao. Discrete cosine transform. *IEEE transactions on Computers*, 100(1):90–93, 1974. 4

[2] Devansh Arpit, Stanisaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *ICML*, 2017. 1

[3] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *BMVC*, 2012. 5

[4] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *CVPR*, 2019. 1, 2, 7, 8

[5] Tao Dai, Hua Zha, Yong Jiang, and Shu-Tao Xia. Image super-resolution via residual block attention networks. In *CVPRW*, 2019. 1, 2

[6] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *TPAMI*, 2016. 1, 2

[7] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *ECCV*, 2016. 1, 2

[8] Yuchen Fan, Jiahui Yu, Yiqun Mei, Yulun Zhang, Yun Fu, Ding Liu, and Thomas S Huang. Neural sparse representation for image restoration. In *NeurIPS*, 2020. 7

[9] Linjing Fang, Fred Monroe, Sammy Weiser Novak, Lyndsey Kirk, Cara R Schiavon, B Yu Seungyoon, Tong Zhang, Melissa Wu, Kyle Kastner, Alaa Abdel Latif, et al. Deep learning-based point-scanning super-resolution imaging. *Nature Methods*, pages 1–11, 2021. 1

[10] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, 2019. 2

[11] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita. Deep back-projection networks for super-resolution. In *CVPR*, 2018. 1, 7, 8

[12] Xiangyu He, Zitao Mo, Peisong Wang, Yang Liu, Mingyuan Yang, and Jian Cheng. Ode-inspired network design for single image super-resolution. In *CVPR*, 2019. 7

[13] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018. 1, 2, 3

[14] Ping Hu, Bing Shuai, Jun Liu, and Gang Wang. Deep level sets for salient object detection. In *CVPR*, 2017. 3

[15] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *CVPR*, 2015. 5

[16] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. In *NeurIPS*, 2016. 3

[17] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, 2016. 1, 2

[18] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014. 5

[19] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *CVPR*, 2017. 7

[20] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017. 1

[21] Jianan Li, Xiaodan Liang, Yunchao Wei, Tingfa Xu, Jiashi Feng, and Shuicheng Yan. Perceptual generative adversarial networks for small object detection. In *CVPR*, 2017. 1

[22] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. Selective kernel networks. In *CVPR*, 2019. 2

[23] Zhen Li, Jinglei Yang, Zheng Liu, Xiaomin Yang, Gwanggil Jeon, and Wei Wu. Feedback network for image super-resolution. In *CVPR*, 2019. 7

[24] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPRW*, 2017. 2, 5, 7, 8

[25] Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S Huang. Non-local recurrent network for image restoration. In *NeurIPS*, 2018. 7

[26] Jie Liu, Wenjie Zhang, Yuting Tang, Jie Tang, and Gangshan Wu. Residual feature aggregation network for image super-resolution. In *CVPR*, 2020. 7, 8

[27] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, 2001. 5

[28] Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. Sketch-based manga retrieval using manga109 dataset. *Multimedia Tools and Applications*, 2017. 5

[29] Yiqun Mei, Yuchen Fan, Yuqian Zhou, Lichao Huang, Thomas S Huang, and Humphrey Shi. Image super-resolution with cross-scale non-local attention and exhaustive self-exemplars mining. In *CVPR*, 2020. 1, 7, 8

[30] Ben Niu, Weilei Wen, Wenqi Ren, Xiangde Zhang, Lianping Yang, Shuzhen Wang, Kaihao Zhang, Xiaochun Cao, and Haifeng Shen. Single image super-resolution via a holistic attention network. In *ECCV*, 2020. 2, 7

[31] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 5

[32] Chang Qiao, Di Li, Yuting Guo, Chong Liu, Tao Jiang, Qionghai Dai, and Dong Li. Evaluation and development of deep neural networks for image super-resolution in optical microscopy. *Nature Methods*, pages 1–9, 2021. 1

[33] Zequn Qin, Pengyi Zhang, Fei Wu, and Xi Li. Fcanet: Frequency channel attention networks. *arXiv preprint arXiv:2012.11879*, 2020. 1, 2, 3, 4, 5

[34] Abhijit Guha Roy, Nassir Navab, and Christian Wachinger. Recalibrating fully convolutional networks with spatial and channel "squeeze and excitation" blocks. *TMI*, 2018. 2

[35] Hang Su, Varun Jampani, Deqing Sun, Orazio Gallo, Erik Learned-Miller, and Jan Kautz. Pixel-adaptive convolutional neural networks. In *CVPR*, 2019. 3

[36] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010. 2

[37] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *ICCV*, 2017. 7

[38] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, Lei Zhang, Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, Kyoung Mu Lee, et al. Ntire 2017 challenge on single image super-resolution: Methods and results. In *CVPRW*, 2017. 5

[39] Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. In *CVPR*, 2020. 2

[40] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *TIP*, 2004. 5

[41] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *ECCV*, 2018. 2

[42] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *CVPR*, 2019. 3

[43] Yuan Yuan, Xiangtao Zheng, and Xiaoqiang Lu. Hyperspectral image superresolution by transfer learning. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(5):1963–1974, 2017. 1

[44] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *Proc. 7th Int. Conf. Curves Surf.*, 2010. 5

[45] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Learning a single convolutional super-resolution network for multiple degradations. In *CVPR*, 2018. 7

[46] Richard Zhang. Making convolutional networks shift-invariant again. In *ICML*, 2019. 3

[47] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, 2018. 1, 2, 3, 5, 7, 8

[48] Yulun Zhang, Kunpeng Li, Kai Li, Bineng Zhong, and Yun Fu. Residual non-local attention networks for image restoration. In *ICLR*, 2019. 7

[49] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *CVPR*, 2018. 1, 2, 7, 8

[50] Shangchen Zhou, Jiawei Zhang, Wangmeng Zuo, and Chen Change Loy. Cross-scale internal graph neural network for image super-resolution. In *NeurIPS*, 2020. 7

[51] Xueyan Zou, Fanyi Xiao, Zhiding Yu, and Yong Jae Lee. Delving deeper into anti-aliasing in convnets. In *BMVC*, 2020. 2, 3