Image Super-Resolution with Non-Local Sparse Attention

Yiqun Mei, Yuchen Fan, Yuqian Zhou
University of Illinois at Urbana-Champaign

Abstract

Both Non-Local (NL) operation and sparse representation are crucial for Single Image Super-Resolution (SISR). In this paper, we investigate their combinations and propose a novel Non-Local Sparse Attention (NLSA) with dynamic sparse attention pattern. NLSA is designed to retain long-range modeling capability from NL operation while enjoying robustness and high-efficiency of sparse representation. Specifically, NLSA rectifies non-local attention with spherical locality sensitive hashing (LSH) that partitions the input space into hash buckets of related features. For every query signal, NLSA assigns a bucket to it and only computes attention within the bucket. The resulting sparse attention prevents the model from attending to locations that are noisy and less-informative, while reducing the computational cost from quadratic to asymptotic linear with respect to the spatial size. Extensive experiments validate the effectiveness and efficiency of NLSA. With a few non-local sparse attention modules, our architecture, called non-local sparse network (NLSN), reaches state-of-the-art performance for SISR quantitatively and qualitatively.

1. introduction

Single Image Super-Resolution (SISR) has attracted great attention in recent years. In general, the goal of SISR is to reconstruct a high-resolution image given its low-resolution counterpart. Due to the ill-posed nature of SISR task, a variety of image priors [12, 14, 24, 36, 40, 46] were proposed as the regularizers, including the most representative sparse and non-local priors, which are the focus of this paper.

For decades, sparsity constraints have been well-explored as a powerful driving forces for many image reconstruction problems [4, 7, 19], especially SISR [46]. With sparse coding, images are well-expressed as the sparse linear combinations of atoms in a predefined over-complete dictionary such as wavelet [11] and curvelet [9] functions. Combining with exemplar-based approaches, sparse representation developed the dictionary using raw image patches [46] or learned semantic feature patches from the degraded image itself [17, 19] or external datasets[47]. As the deep Convolution Neural Networks (CNNs) for SISR emerges, the non-linearity activation between layers embraces the benefits of sparsity prior. Dong *et al*. propose the SRCNN [16] to first successfully bridge convolution to classic sparse coding where the ReLU activation roughly enforced 50% sparsity by zeroing-out all negative entries. Recently, Fan *et al*. [21] go beyond that by explicitly imposing sparsity constraints upon hidden neurons and conclude that sparsity in feature representation is indeed beneficial and favorable. It is widely proven that the sparsity constraints lead to high efficiency by largely decreasing the number of elements to represent images. It also yields a more powerful and robust expression in handling inverse problems theoretically [10, 20] and practically.

Another widely-explored image prior is the Non-Local (NL) prior. For SISR, adopting Non-Local Attention becomes a more prevalent way [37, 51] to utilize the image self-similarity prior that small patterns tend to recur within the same image [5]. NL operation searches for those similar patterns globally, and selectively sums over those correlated features to enhance the representation. Though Non-Local Attention is intuitive and promising to fuse features, directly applying it for SISR task will encounter some issues that cannot be ignored. First, the receptive field of features in deeper layers tend to be global, thus the mutual-correlation computation among deep features are not that accurate [33]. Second, global NL attention requires the computation of feature mutual-similarity among all the pixel locations. It results in quadratic computational cost with respect to image size. To alleviate the above mentioned problems, one strategy is to limit the NL searching range within a local neighbourhood. But it reduces computational cost at the expense of missing much global information.

In this paper, for the specific SISR task, we aim to enforce sparsity in the Non-Local attention module, as well as largely reduce its computational cost. Specifically, we propose a novel *Non-Local Sparse Attention (NLSA)* and embed it into a residual network baseline like EDSR [32] to form a *Non-Local Sparse Network (NLSN)*. To force the sparsity of the NLSA, we spatially partition the deep feature pixels into different groups (termed *attention buckets* in this paper). The feature pixels inside the same bucket are considered content closely-correlated. We then apply the Non-Local (NL) operation within the bucket that the query pixel belongs to, or across adjacent buckets after sorting. We achieve this by building the partition approach upon Locality Sensitive Hashing (LSH) research [23] that searches for similar elements which produce maximum inner product.

The proposed NLSA will make it possible to reduce the computational complexity of NL from quadratic to asymptotic linear with respect to spatial dimensions. Searching similar cues within a smaller content-correlated bucket will also make the module attend to locations which are more informative and related. As a result, NLSA retains global modeling ability of the standard NL operation, while enjoying robustness and efficiency from its sparse representation. In summary, the main contributions of our paper are:

- We propose to enforce sparsity in Non-Local operation for SISR task via a novel Non-Local Sparse Attention (NLSA) module. The sparsity constraint forces the module to focus on correlated and informative area while ignoring unrelated and noisy contents.
- We achieve the feature sparsity by first grouping the feature pixels and only conducting Non-Local operations within the group named attention bucket. We adopt the Locality Sensitive Hashing (LSH) for grouping and assign each group a Hash code. The proposed approach significantly reduces the computational complexity from quadratic to asymptotic linear.
- Without any bells and whistles, a few NLSA modules can drive a fairly simple ResNet backbone to state-of-the-arts. Extensive experiments demonstrate the advantages of NLSA over the standard Non-Local Attention (NLA).

2. Related Work

2.1. Sparse representation.

In this section, we briefly review the key concepts of the sparse representation. Formally, Suppose $x_1, x_2, ..., x_n \in R^d$ are n known examples in an over-complete dictionary $D^{d\times n}$ ($d < n$). For a query signal $y \in R^d$, exemplar-based approaches [12, 46] represent it as a weighted sum of the elements in D:

$$y = \alpha_1 x_1 + \alpha_2 x_2 + ... + \alpha_n x_n \tag{1}$$

where α_i is a coefficient with respect to the x_i. Let $\alpha = [\alpha_1, \alpha_2, ..., \alpha_n]$, then Eq. 1 can be written as:

$$y = D\alpha \tag{2}$$

Eq. 2 yields an undetermined linear system. Solving α becomes an ill-posed problem. To alleviate this, sparse representation assumes that y should be sparsely represented, i.e., α should be sparse:

$$y = D\alpha, \;\; s.t. \; \|\alpha\|_0 \le k \tag{3}$$

Where $\|.\|_0$ and k counts and bounds the number of non-zero elements in α, respectively. Given the sparsity constraint, optimization methods like OMP [38] can effectively approximate the solution of Eq 3. The resulting sparse representation has been proven to be extremely powerful in the field of image reconstruction [19, 47, 53]. Motivated by their success, we are inspired to incorporate sparse representation into non-local attention.

2.2. Non-Local Attention (NLA) for image SR.

Non-local operation assumes that small patches tend to re-occur within the same image, which has been well-demonstrated to be a strong prior for natural images [5]. Non-local approaches were designed to utilize these self-recurrences to recover underlying signals. Non-local operation has been widely applied in many image restoration problems, such as super-resolution [22], denoising [1, 2, 8, 13], and inpainting [18]. Wang *et al.* [45] first bridges classic non-local filtering to self-attention methods [43] for machine translation and further introduces Non-Local Attention (NLA) into deep neural networks to capture global semantic relationships for high-level tasks. For image super-resolution, recent approaches, such as NLRN [33], SAN [15], RNAN [51], and CSNLN [37], demonstrate considerable benefits of exploring long-range feature correlations by adopting NL attention. However, the existing NLAs designed for SISR task are either limited to a local neighbourhood, or largely computational resources-consuming. Motivated by recent progress [29, 39, 44] on self-attention methods for language modeling, we propose Non-Local Sparse Attention (NLSA) to embrace the long-range information as well as reducing the complexity.

3. Non-Local Sparse Attention (NLSA)

3.1. General Form of Sparse Attention

As discussed above, the merits of Non-Local Attention for image SR often come at a price of limiting its searching range. To alleviate the issue, we propose to bridge standard NLA to exemplar-based approaches and then break the tie by imposing sparsity constraint.

Non-Local Attention. In general, a Non-Local Attention enhances an input feature map $X \in R^{h\times w\times c}$ by summarizing information from all positions. For illustration purpose, we reshape X into an 1-D feature $X \in R^{n\times c}$ where $n = hw$. Given a query location i, the corresponding

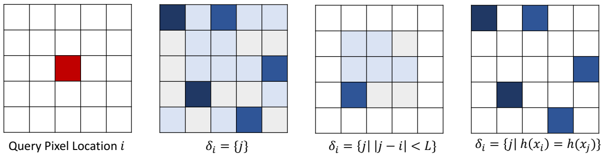

Query Pixel Location i
$\delta_i = \{j\}$
$\delta_i = \{j|\ |j - i| < L\}$
$\delta_i = \{j|\ h(x_i) = h(x_j)\}$

Figure 1. Examples of the Attention Bucket in 2D spatial space. Given a query at location i and an index set δ_i, where the δ_i decides on the group of locations to compute the non-locally fused features. Darker blue regions in the figure form the attention bucket. $\delta_i = \{j\}$ indicates using the full-range pixels like in the standard NL attention. $\delta_i = \{j \,|\, |j - i] < L\}$ indicates a local neighbourhood constrained attention span. While $\delta_i = \{j \,| h(x_i) = h(x_j)\}$ is the proposed hash-based attention bucket.

output response $y_i \in R^c$ can be expressed as:

$$y_i = \sum_{j=1}^{n} \frac{f(x_i, x_j)}{\sum_{\hat{j}=1}^{n} f(x_i, x_{\hat{j}})} g(x_j) \tag{4}$$

where x_i ,x_j and $x_{\hat{j}}$ are pixel-wise features at location i, j and $\hat{j}$ on X. $f(.,.)$ measures the mutual similarity and $g(.)$ is a feature transformation function. Eq. 4 can be viewed as an exemplar-based approach (Eq. 2) by setting $D = [g(x_1), ..., g(x_n)] \in R^{c\times n}$ and $\alpha_i = [f(x_i, x_1), ..., f(x_i, x_n)] \in R^n$, i.e., $y_i = D\alpha_i$.

Sparsity Constraints on Non-Local Attention. Given Eq. 4, a sparsity constraint can be imposed on the Non-Local Attention, by limiting the number of non-zero entries of α up to a constant k. Therefore, a general form of Non-Local Attention with Sparse Constraints can be derived as:

$$y_i = D\alpha_i \quad \text{s.t.} \quad \|\alpha_i\|_0 \leq k \tag{5}$$

$$= \sum_{j\in\delta_i} \frac{f(x_i, x_j)}{\sum_{\hat{j}\in\delta_i} f(x_i, x_{\hat{j}})} g(x_j) \tag{6}$$

where δ_i indexes non-zero elements of α_i, i.e., $\delta_i = \{j \,|\, \alpha_i[j] \neq 0\}$, where $\alpha_i[j]$ denotes the j-th element in α_i. With sparse attention, the computational cost can be largely saved by ignoring elements with zero coefficients.

Attention Bucket. Notice that the index set δ_i indicates the group of pixel locations where a given query should attend to. In another word, δ_i constrains the identified locations where the Non-Local Attention can be computed from. In this paper, we define this group of locations as in an *Attention Bucket*. Figure 1 shows some examples of the attention bucket under different δ_i. For example, standard non-local attention spans over all the possible locations, which makes the aggregated feature noisy and less informative. If an attention spans a local neighborhood of length L, this specifies a window $\delta_i = \{j \,|\, |j - i| < L\}$. In this case, some long-range context cannot be effectively aggregated.

Intuitively, a more powerful sparse attention is expected to cover locations that are the most informative and closely related at a global scale, resulting that ignoring other elements brings no harm to the performance. A naïve way is to rank all mutual similarities and then use the top k entries. However, it requires forming a full attention first which brings no efficiency improvement. In the following sections, we will show how we form the attention bucket for each query i by globally modelling the attention with high efficiency.

3.2. Attention Bucket from Locality Sensitive Hashing (LSH)

As discussed above, a desired attention should not only keep sparse but also incorporate the most relevant elements. In this section, we propose to adopt the Spherical Locality Sensitive Hashing (LSH) [3, 42] to form the desired attention bucket containing global and correlated elements with the query element. Specifically, we propose to spatially partition the embedding space into buckets of similar features depending on their angular distances. Consequently, even when the attention keeps sparse by only spanning over one bucket, it could still capture most of the correlated elements.

Recall that a hashing scheme is locality sensitive if nearby elements are at high possibility to fall into the same hash bucket (hash code) whereas distant ones are not. The spherical LSH is an instance of LSH designed for angular distance. One can intuitively think it as randomly rotating a cross-polytope inscribed into a hyper-sphere, as shown in the top branch of Figure 2. The hash function projects a tensor onto the hyper-sphere and the closest polytope vertex is selected as its hash code. Thus, if two vectors have a small angular distance, they are likely to fall in the same hash bucket, which is also the defined attention bucket.

Formally, suppose we want to get m hash buckets, we have to first project the target tensor onto a hyper-sphere and randomly rotate it with a matrix $A \in R^{c\times m}$, a sampled random rotation matrix with i.i.d. Gaussian entries, i.e.,

$$\hat{x} = A(\frac{x}{\|x\|_2}) \tag{7}$$

The hash code, or the assigned hush bucket, is defined as $h(x) = \arg\max_i(\hat{x})$. After hashing all elements, we manage to partition the space into buckets of correlated elements, and the attention bucket of x_i can be identified by the index set $\delta_i = \{j|h(x_j) = h(x_i)\}$.

In practice, the spherical LSH is simultaneously performed for all elements with batch matrix multiplication, which only adds negligible computational cost. Knowing which bucket to attend in advance, the model can achieve high-efficiency and robustness by ignoring other noisy or less-correlated partitions.

3.3. Non-Local Sparse Attention

Once the attention bucket index set δ_i for query location i is determined, the proposed Non-Local Sparse Attention

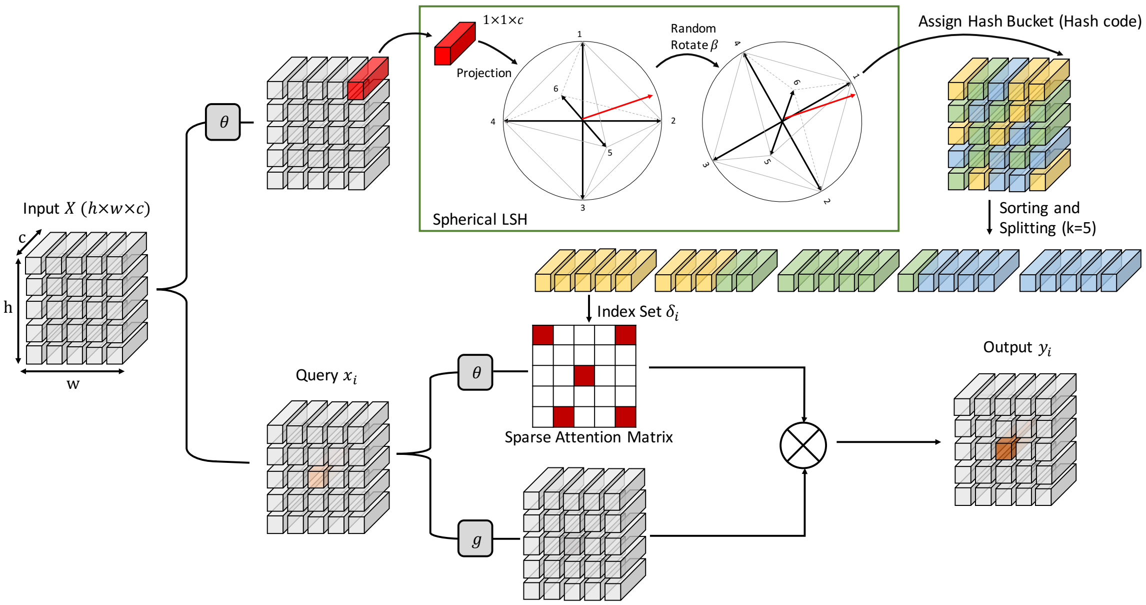


Figure 2. The proposed Non-Local Sparse Attention (NLSA). The upper branch partitions the input features into buckets via Spherical Locality Sensitive Hashing (LSH). The bottom branch computes attention for every query within each bucket or between adjacent buckets after sorting the buckets by the hash code. The module embraces the advantages of Non-Local Attention (modeling globally), and the benefits of sparsity and hashing (high efficiency).

(NLSA) can be easily derived from Eq 6. Specifically, as shown in Figure 2, NLSA assigns each pixel-wise feature in X to a bucket sharing the same hash code based on their content relevance, and only the corresponding bucket elements contribute to the output. In the following, we describe some techniques used in our practical implementation.

Dealing with Unbalanced Bucketing. Ideally, given totally m buckets, each hash bucket will equally contain $\frac{n}{m}$ elements. However, this may not hold in practice as buckets tend to be unbalanced. This also makes parallel computing very difficult. To overcome the difficulty, we first sort features by their bucket value (hash code), then the permutation is defined as $\pi : i \rightarrow \pi(i)$. After knowing their new positions (denoted by the superscripts), we split them into chunks of size k:

$$C_j = [x^{jk+1}, x^{jk+2}, ..., x^{(j+1)k}] \tag{8}$$

where C_j presents the j-th chunk. Consequently, the attention bucket of x_i is updated to the corresponding chunk,

$$\delta_i = Index(C_j) \qquad \text{if } \pi(i) \in [jk+1, (j+1)k] \tag{9}$$

The above strategy is more friendly used to perform computation in parallel. Despite its merits, splitting the original buckets into fixed-size chunks as the updated attention buckets also brings a subtle issue: some new chunks may cross the original bucket boundaries, as shown in Figure 2. Fortunately, this issue can be effectively alleviated by allowing attention to also span over adjacent chunks.

Multi-round NLSA. The nature of Spherical LSH indicates there is always a small chance that some correlated elements are incorrectly hashed into different hash buckets. Fortunately, this chance can be reduced by independently hashing multiple rounds and taking the union of all results. Motivated by this observation, we propose multi-rounds NLSA to make hashing process more robust. Let $\delta_{r,i}$ denote the resulting attention bucket of x_i of the r-th hashing, and $Att(x_i, \delta_{r,i})$ be the associated sparse attention defined in Eq. 6, i.e.,

$$Att(x_i, \delta_{r,i}) = \sum_{j \in \delta_{r,i}} \frac{f(x_i, x_j)}{\sum_{\hat{j} \in \delta_{r,i}} f(x_i, x_{\hat{j}})} g(x_j) \tag{10}$$

Then the multi-round NLSA is defined as:

$$x_i = \sum_r \frac{\sum_{j \in \delta_{r,i}} f(x_i, x_j)}{\sum_{\hat{r}} \sum_{\hat{j} \in \delta_{\hat{r},i}} f(x_i, x_{\hat{j}})} Att(x_i, \delta_{r,i}) \tag{11}$$

Intuitively, multi-round NLSA is the weighted sum of each single round attention results, and the weight coefficient represents the normalized similarity between the query and the elements in its assigned hash bucket for each round. As a side effect, this augmentation linearly increases computational cost with respect to the total hash rounds. But we can still dynamically adjust this parameter during the evaluation time to study the trade-offs.

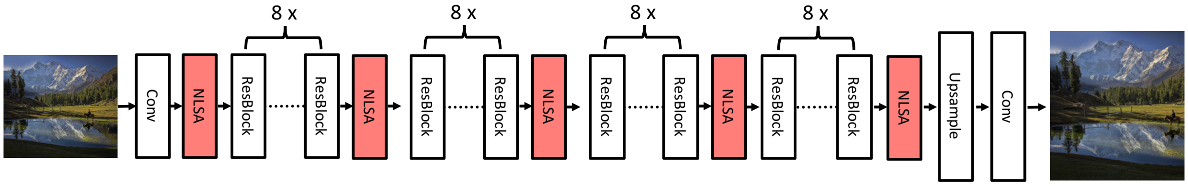


Figure 3. The proposed non-local sparse network (NLSN). Five sparse attention modules are inserted after every 8 residual blocks.

Computational complexity. We analyze the time complexity of the proposed NLSA. Given an input feature $X \in R^{n\times c}$, the cost for spherical LSH with m buckets a matrix multiplication, which is $\mathcal{O}(ncm)$. The cost for attention operation (Eq. 6) with sparsity constraint (size of the attention bucket) k is $\mathcal{O}(nck)$. The sorting operation of a sequence with length n and m distinct numbers (bucket number) adds an additional $\mathcal{O}(nm)$ with quick sort (and can be further optimized with advanced sorting algorithms). Therefore, the overall computational cost of our non-local sparse attention is $\mathcal{O}(nck + ncm + nm)$. Hashing for r rounds will increase computational cost with a factor r, resulting in $\mathcal{O}(rnck + rncm + rnm)$. NLSA only takes linear computational complexity with respect to the input spatial size.

Instantiations. To instantiate the non-local attention defined in Eq. 6, we choose embedded Gaussian for $f(.,.)$, i.e. $f(x_i, x_j) = \exp(\theta(x_i)^T\phi(x_j))$, where θ and ϕ are learned linear projections. In this paper, we use one of its variants that sets $\theta = \phi$ to ensure the projected features are in the same subspace for better LSH. We also found sharing θ and ϕ did not hurt the performance, which will be verified in experiment section.

3.4. Non-Local Sparse Network (NLSN)

To demonstrate the effectiveness of the non-local sparse attention, we build our non-local sparse network (NLSN) upon a fairly simple EDSR [32] backbone, which consists of 32 residual blocks. As shown in Figure 3, the network uses total 5 attention blocks with one insertion after every 8 residual blocks. The network is trained solely with L_1 reconstruction loss.

4. Experiments

4.1. Datasets and Metrics

Following [32, 52], we use DIV2K as our training dataset, which contains 800 training images. We test our approach on 5 standard benchmarks: Set5 [6], Set14 [48], B100 [34], Urban100 [27] and Manga109 [35]. We evaluate all the results using PSNR and SSIM metrics on Y channel in the transformed YCbCr space.

4.2. Implementation and Training Details

For the non-local sparse attention, we set the attention bucket size (i.e. chunk size) $k = 144$. The corresponding number of hash buckets $m = \min(\frac{hw}{k}, 128)$ is dynamically determined by the division of input size $h \times w$ and k but clipped at 128. The final non-local sparse network is built upon an EDSR backbone with 32-residual blocks and 5 additional NLSA blocks. We set all the convolutional kernel sizes to 3×3. All intermediate features have 256 channels the same as in EDSR, except for those embedded ones in the attention blocks, which have 64 channels. The last convolution layer transforms the deep features into a n 3-channel RGB image with 3 filters. By default, the model is trained and evaluated with NLSA of $r = 4$ rounds.

During training, we randomly crop 48×48 patches from the training examples and form a mini-batch of 16 images. The training images are further augmented via horizontal flipping and random rotation of 90, 180, and 270 degrees. We optimize the model by ADAM optimizer [28] with $\beta_1 = 0.9$, $\beta_2 = 0.99$ and $\epsilon = 10^{-8}$. The learning rate is set to 10^{-4} and reduced by 0.5 after 200 epochs. The final model is obtained after 1000 epochs. Our model is implemented using PyTorch and trained on Nvdia 1080ti GPUs.

4.3. Comparisons with State-of-the-Arts

To demonstrate the effectiveness of our NLSA, we compare it with 12 state-of-the-arts including LapSRN [30], SRMDNF [49], MemNet [41], EDSR [32], DBPN [25], RDN [52], RCAN [50], NLRN[33], RNAN [51], SRFBN [31], OISR [26] and SAN [15].

The quantitative results are shown in Table 1. Our NLSN achieves the best results on almost all benchmarks and all upsampling scales. In particular, when comparing with its backbone EDSR, adding additional NLSAs shows great superiority in improving performance and even makes EDSR outperform the very competitive RCAN and SAN. Especially, the proposed NLSN brings improvements around 0.2 dB in Set5 and Set14, 0.1 dB in B100 and more than 0.4 dB in Urban100 and Manga109. These performance gains show that NLSA succeeds in exploring extensive global cues for more accurate super-resolution. Moreover, when comparing with previous non-local approaches like NLRN and RNAN, our network shows a huge advance in all en-

Table 1. Quantitative results on benchmark datasets. Best and second best results are **highlighted** and underlined.

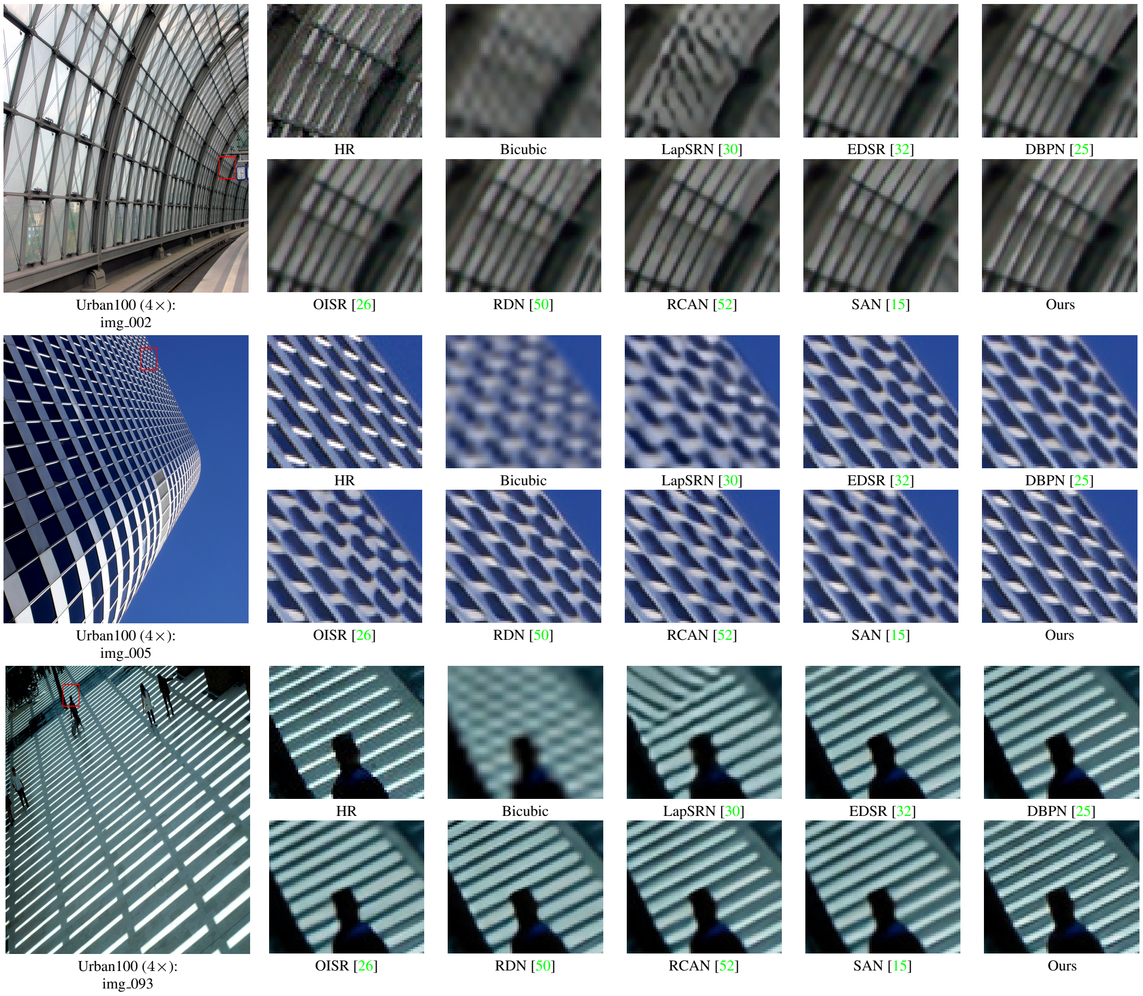
Method	Scale	Set5		Set14		B100		Urban100		Manga109	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
LapSRN [30]	×2	37.52	0.9591	33.08	0.9130	31.08	0.8950	30.41	0.9101	37.27	0.9740
MemNet [41]	×2	37.78	0.9597	33.28	0.9142	32.08	0.8978	31.31	0.9195	37.72	0.9740
SRMDNF [49]	×2	37.79	0.9601	33.32	0.9159	32.05	0.8985	31.33	0.9204	38.07	0.9761
DBPN [25]	×2	38.09	0.9600	33.85	0.9190	32.27	0.9000	32.55	0.9324	38.89	0.9775
RDN [52]	×2	38.24	0.9614	34.01	0.9212	32.34	0.9017	32.89	0.9353	39.18	0.9780
RCAN [50]	×2	38.27	0.9614	**34.12**	0.9216	32.41	0.9027	33.34	0.9384	39.44	0.9786
NLRN [33]	×2	38.00	0.9603	33.46	0.9159	32.19	0.8992	31.81	0.9249	–	–
RNAN [51]	×2	38.17	0.9611	33.87	0.9207	32.32	0.9014	32.73	0.9340	39.23	0.9785
SRFBN [31]	×2	38.11	0.9609	33.82	0.9196	32.29	0.9010	32.62	0.9328	39.08	0.9779
OISR [26]	×2	38.21	0.9612	33.94	0.9206	32.36	0.9019	33.03	0.9365	–	–
SAN [15]	×2	38.31	**0.9620**	34.07	0.9213	32.42	**0.9028**	33.10	0.9370	39.32	**0.9792**
EDSR [32]	×2	38.11	0.9602	33.92	0.9195	32.32	0.9013	32.93	0.9351	39.10	0.9773
NLSN (ours)	×2	**38.34**	0.9618	34.08	**0.9231**	**32.43**	0.9027	**33.42**	**0.9394**	**39.59**	0.9789
LapSRN [30]	×3	33.82	0.9227	29.87	0.8320	28.82	0.7980	27.07	0.8280	32.21	0.9350
MemNet [41]	×3	34.09	0.9248	30.00	0.8350	28.96	0.8001	27.56	0.8376	32.51	0.9369
SRMDNF [49]	×3	34.12	0.9254	30.04	0.8382	28.97	0.8025	27.57	0.8398	33.00	0.9403
RDN [52]	×3	34.71	0.9296	30.57	0.8468	29.26	0.8093	28.80	0.8653	34.13	0.9484
RCAN [50]	×3	34.74	0.9299	30.65	0.8482	29.32	0.8111	29.09	0.8702	34.44	0.9499
NLRN [33]	×3	34.27	0.9266	30.16	0.8374	29.06	0.8026	27.93	0.8453	-	-
RNAN [51]	×3	34.66	0.9290	30.52	0.8462	29.26	0.8090	28.75	0.8646	34.25	0.9483
SRFBN [31]	×3	34.70	0.9292	30.51	0.8461	29.24	0.8084	28.73	0.8641	34.18	0.9481
OISR [26]	×3	34.72	0.9297	30.57	0.8470	29.29	0.8103	28.95	0.8680	-	-
SAN [15]	×3	34.75	0.9300	30.59	0.8476	29.33	0.8112	28.93	0.8671	34.30	0.9494
EDSR [32]	×3	34.65	0.9280	30.52	0.8462	29.25	0.8093	28.80	0.8653	34.17	0.9476
NLSN (ours)	×3	**34.85**	**0.9306**	**30.70**	**0.8485**	**29.34**	**0.8117**	**29.25**	**0.8726**	**34.57**	**0.9508**
LapSRN [30]	×4	31.54	0.8850	28.19	0.7720	27.32	0.7270	25.21	0.7560	29.09	0.8900
MemNet [41]	×4	31.74	0.8893	28.26	0.7723	27.40	0.7281	25.50	0.7630	29.42	0.8942
SRMDNF [49]	×4	31.96	0.8925	28.35	0.7787	27.49	0.7337	25.68	0.7731	30.09	0.9024
DBPN [25]	×4	32.47	0.8980	28.82	0.7860	27.72	0.7400	26.38	0.7946	30.91	0.9137
RDN [52]	×4	32.47	0.8990	28.81	0.7871	27.72	0.7419	26.61	0.8028	31.00	0.9151
RCAN [50]	×4	32.63	0.9002	28.87	0.7889	27.77	0.7436	26.82	0.8087	31.22	0.9173
NLRN [33]	×4	31.92	0.8916	28.36	0.7745	27.48	0.7306	25.79	0.7729	-	-
RNAN [51]	×4	32.49	0.8982	28.83	0.7878	27.72	0.7421	26.61	0.8023	31.09	0.9149
SRFBN [31]	×4	32.47	0.8983	28.81	0.7868	27.72	0.7409	26.60	0.8015	31.15	0.9160
OISR [26]	×4	32.53	0.8992	28.86	0.7878	27.75	0.7428	26.79	0.8068	-	-
SAN [15]	×4	**32.64**	**0.9003**	**28.92**	0.7888	**27.78**	0.7436	26.79	0.8068	31.18	0.9169
EDSR [32]	×4	32.46	0.8968	28.80	0.7876	27.71	0.7420	26.64	0.8033	31.02	0.9148
NLSN (ours)	×4	32.59	0.9000	28.87	**0.7891**	**27.78**	**0.7444**	**26.96**	**0.8109**	**31.27**	**0.9184**

tries. This is mainly because NLSA only attends to content-correlated locations, which yields a more accurate correlation estimation. It is worth noting that all these benefits come only at an expense of adding a small amount of computation, which roughly equals to a few convolution operations, demonstrating that our NLSA is indeed effective and efficient. Visual results on Urban100 are shown in Figure 4. Our NLSN effectively restores the image details by efficiently utilizing global similar patches.

4.4. Ablation Study

In this section, we conduct controlled experiments to analyze the proposed NLSA. We build the baseline model with 16 residual blocks. For each attention variants, we insert a corresponding block after the 8-th residual block.

Size k of the Attention Bucket. As discussed above, the sparsity of NLSA is controlled by the size of the attention bucket k (chunk size). And if most correlated elements are successfully identified, a small k should be sufficient for producing high quality super resolution. Here we investigate the effects of different k and compare it with standard Non-Local Attention that have local window receptive fields covering exactly k elements. Specifically, we set $k = \{5^2, 10^2, 15^2, 20^2, 25^2, 30^2, 40^2, 50^2\}$. As shown in Figure 5, the performance of NLSA peaks at $k = 10^2$, which significantly outperforms the NLA of the same cov-



Urban100 (4×): img_002

HR Bicubic LapSRN [30] EDSR [32] DBPN [25]

OISR [26] RDN [50] RCAN [52] SAN [15] Ours

Urban100 (4×): img_005

HR Bicubic LapSRN [30] EDSR [32] DBPN [25]

OISR [26] RDN [50] RCAN [52] SAN [15] Ours

Urban100 (4×): img_093

HR Bicubic LapSRN [30] EDSR [32] DBPN [25]

OISR [26] RDN [50] RCAN [52] SAN [15] Ours

Figure 4. Visual comparison for 4× SR on Urban100 dataset

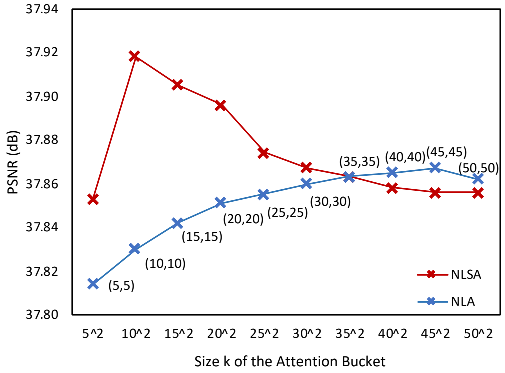


Figure 5. Ablation study on size k of the attention bucket.

erage about 0.1 dB. Moreover, attending to these 100 locations even yields much better results than the best overall performance of NL attention, which attends to more than 2000 (45^2) locations. These results demonstrate our NLSA manages to identify the most informative positions at a global scale. This also indicates that knowing where to attend is more important than attending more.

Table 2. Comparison of partition strategies on Set5 (×2).

Parition	Baseline	Local(q=12)	Random	Spherical LSH
PSNR	37.78	37.83	37.79	37.92

When further enlarging the attention bucket, NLSA becomes much denser. Its performance also begins to approximate the standard NLA as expected. This is mainly because a larger k reduces the effectiveness of LSH by decreasing the number of hash bits and also is more likely to make chunks across multiple bucket boundaries. In an extreme case when k equals to the number of image pixels, NLSA will be just identical to the standard Non-Local Attention.

Similarity Grouping via Spherical LSH. Spherical LSH partitions the space into attention buckets of correlated elements and it plays a key role in NLSA. We investigate its effectiveness by comparing it with other partition options. We first compare it with random hashing, i.e., elements are assigned to random buckets. As shown in Table 2, random hashing brings no evident improvement over the baseline as expected. In contrast, Spherical LSH achieves more than 0.1 dB improvements over the random version and baseline.

Table 3. Ablation study on the attention rounds on Set5 (×2). r denotes the number of attention rounds.

train \ test	$r=1$	$r=2$	$r=4$	$r=8$
$r=1$	37.86	37.87	37.88	37.89
$r=2$	31.87	31.88	31.89	37.90
$r=4$	37.87	37.90	37.92	37.93
$r=8$	37.88	37.90	37.92	37.94

Table 4. Effect of sharing linear projection on Set5 (×2).

Linear Projection	Baseline	$\theta \neq \phi$	$\theta = \phi$
PSNR	37.78	37.86	37.87

We also implement the local window strategy, where elements are gathered purely depending on locations. We set the window size $q = 12$, which equals to the attention range ($k = 144$) for fair comparison. Table 2 shows Spherical LSH is superior than local window strategy. This indicates Spherical LSH indeed effectively identifies more useful global cues beyond a local neighborhood.

Multi-round NLSA As discussed above, hashing for more rounds improves the robustness of NLSA but at a price of linearly increasing computational cost. Fortunately, the attention rounds r can be flexibly adjusted during testing. Results of the models trained and evaluated with different rounds are presented in Table 3. The results indicate that increasing the hashing rounds at either training or evaluation can constantly improve super-resolution accuracy. As expected, the best result is achieved with the largest round number during both training and testing, but it also yields the worst computational cost.

Shared linear projection in embedded Gaussian. NLSA uses shared linear projections in embedded Gaussian to estimate pair-wise similarity, i.e., $\theta = \phi$. We investigate its effect on a standard Non-Local Attention. As shown in Table 4, the model with shared linear projection produces comparable and even slightly better results than the one not shared. In other words, this modification does not bring any harm to the performance.

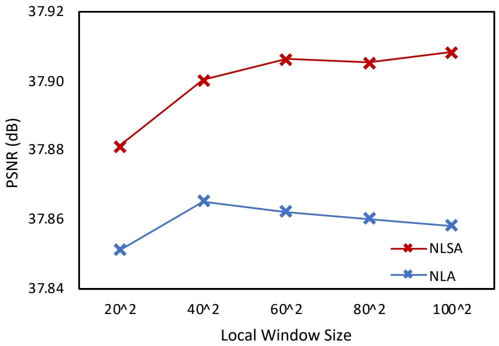


Figure 6. Effect of size of receptive fields on Set5 (×2).

Robustness. Unlike standard Non-Local Attention, our NLSA is inherently robust to receptive fields, since it avoids attending to less-informative and less-correlated locations. As shown in Figure 6, both NLSA and NLA are confined to local windows. It suggests that, for a same window size, our NLSA constantly outperforms NLA. This proves that NLSA captures more accurate correlations and is more robust to noisy information. Further increasing the receptive fields also slightly improves NLSA as it can capture additional information from longer-range contexts. In contrast, the performance of Non-Local Attention gradually drops as more information is taken into account, which hurts its correlation estimation.

Table 5. Efficiency and performance comparison on Set5 (×2). r denotes attention rounds.

Methods	GFLOPs	PSNR
Baseline	0	37.78
NLA	16.0	37.86
Conv	0.7	–
NLSA-r1	0.9	37.87
NLSA-r2	1.4	37.90
NLSA-r4	2.4	37.92
NLSA-r8	4.3	37.93

Efficiency. We compare the NLSA with standard Non-Local Attention in terms of computational efficiency. Table 5 reports the incremental computational cost and the associated performances. The input is assumed to have spatial size of 100 × 100 and both input and output channels are set to 64. We also add an entry of normal 3×3 convolution operation for better illustration. As shown in Table 5, NLSA significantly reduces the computational cost of the NLA while obtaining superior performance. For example, the most efficient single round NLSA-r1 has a similar computational cost of convolution, but achieves comparable performances with standard NL operation. The best result is achieved by NLSA-r8 with 8 rounds attention, which is still roughly 3 times more efficient than the standard Non-Local Attention. More comparisons of the efficiency are provided in the supplementary material.

5. Conclusion

In this paper, we propose a novel Non-Local Sparse Attention (NLSA) for deep single image super resolution networks, that simultaneously embraces the benefits of sparse representation and non-local operation. NLSA globally identifies the most informative locations to attend without paying any attention to unrelated regions, resulting in a robust and efficient global modeling operation. Further inserting it into deep networks, our non-local sparse network sets new state-of-the-arts on multiple benchmarks. Extensive evaluations suggest that our NLSA is a superior operation over the standard non-local attention and indeed beneficial for accurate image super resolution.

References

[1] Abdelrahman Abdelhamed, Mahmoud Afifi, Radu Timofte, and Michael S Brown. Ntire 2020 challenge on real image denoising: Dataset, methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 496–497, 2020. 2

[2] Abdelrahman Abdelhamed, Radu Timofte, and Michael S Brown. Ntire 2019 challenge on real image denoising: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 2

[3] Alexandr Andoni, Piotr Indyk, Thijs Laarhoven, Ilya Razenshteyn, and Ludwig Schmidt. Practical and optimal lsh for angular distance. In *Advances in neural information processing systems*, pages 1225–1233, 2015. 3

[4] Chenglong Bao, Jian-Feng Cai, and Hui Ji. Fast sparsity-based orthogonal dictionary learning for image restoration. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3384–3391, 2013. 1

[5] Michael F Barnsley. *Fractals everywhere*. Academic press, 1998. 1, 2

[6] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *Proceedings of the British Machine Vision Conference*, 2012. 5

[7] José M Bioucas-Dias and Mário AT Figueiredo. A new twist: Two-step iterative shrinkage/thresholding algorithms for image restoration. *IEEE Transactions on Image processing*, 16(12):2992–3004, 2007. 1

[8] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 60–65. IEEE, 2005. 2

[9] Emmanuel J Candes and David L Donoho. Recovering edges in ill-posed inverse problems: Optimality of curvelet frames. *Annals of statistics*, pages 784–842, 2002. 1

[10] Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 52(2):489–509, 2006. 1

[11] Antonin Chambolle, Ronald A De Vore, Nam-Yong Lee, and Bradley J Lucier. Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage. *IEEE Transactions on Image Processing*, 7(3):319–335, 1998. 1

[12] Hong Chang, Dit-Yan Yeung, and Yimin Xiong. Super-resolution through neighbor embedding. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I. IEEE, 2004. 1, 2

[13] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007. 2

[14] Shengyang Dai, Mei Han, Wei Xu, Ying Wu, and Yihong Gong. Soft edge smoothness prior for alpha channel super resolution. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 1

[15] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11065–11074, 2019. 2, 5, 6, 7

[16] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015. 1

[17] Weisheng Dong, Lei Zhang, Guangming Shi, and Xin Li. Nonlocally centralized sparse representation for image restoration. *IEEE transactions on Image Processing*, 22(4):1620–1630, 2012. 1

[18] Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1033–1038. IEEE, 1999. 2

[19] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745, 2006. 1, 2

[20] Michael Elad, Mario AT Figueiredo, and Yi Ma. On the role of sparse and redundant representations in image processing. *Proceedings of the IEEE*, 98(6):972–982, 2010. 1

[21] Yuchen Fan, Jiahui Yu, Yiqun Mei, Yulun Zhang, Yun Fu, Ding Liu, and Thomas S Huang. Neural sparse representation for image restoration. *arXiv preprint arXiv:2006.04357*, 2020. 1

[22] Gilad Freedman and Raanan Fattal. Image and video upscaling from local self-examples. *ACM Transactions on Graphics (TOG)*, 30(2):1–11, 2011. 2

[23] Aristides Gionis et al. Similarity search in high dimensions via hashing. 2

[24] Daniel Glasner, Shai Bagon, and Michal Irani. Super-resolution from a single image. In *2009 IEEE 12th international conference on computer vision*, pages 349–356. IEEE. 1

[25] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Deep back-projection networks for super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1664–1673, 2018. 5, 6, 7

[26] Xiangyu He, Zitao Mo, Peisong Wang, Yang Liu, Mingyuan Yang, and Jian Cheng. Ode-inspired network design for single image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1732–1741, 2019. 5, 6, 7

[27] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5197–5206, 2015. 5

[28] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[29] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2019. 2

[30] 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. 5, 6, 7

[31] Zhen Li, Jinglei Yang, Zheng Liu, Xiaomin Yang, Gwanggil Jeon, and Wei Wu. Feedback network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3867–3876, 2019. 5, 6

[32] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017. 1, 5, 6, 7

[33] Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S Huang. Non-local recurrent network for image restoration. In *Advances in Neural Information Processing Systems*, pages 1673–1682, 2018. 1, 2, 5, 6

[34] 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

[35] 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*, 76(20):21811–21838, 2017. 5

[36] Yiqun Mei, Yuchen Fan, Yulun Zhang, Jiahui Yu, Yuqian Zhou, Ding Liu, Yun Fu, Thomas S Huang, and Honghui Shi. Pyramid attention networks for image restoration. *arXiv preprint arXiv:2004.13824*, 2020. 1

[37] Yiqun Mei, Yuchen Fan, Yuqian Zhou, Lichao Huang, Thomas S Huang, and Honghui Shi. Image super-resolution with cross-scale non-local attention and exhaustive self-exemplars mining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5690–5699, 2020. 1, 2

[38] Yagyensh Chandra Pati, Ramin Rezaiifar, and Perinkulam Sambamurthy Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar conference on signals, systems and computers*, pages 40–44. IEEE, 1993. 2

[39] Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 9:53–68, 2021. 2

[40] Jian Sun, Zongben Xu, and Heung-Yeung Shum. Image super-resolution using gradient profile prior. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 1

[41] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *ICCV*, 2017. 5, 6

[42] Kengo Terasawa and Yuzuru Tanaka. Spherical lsh for approximate nearest neighbor search on unit hypersphere. In *Workshop on Algorithms and Data Structures*, pages 27–38. Springer, 2007. 3

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017. 2

[44] Apoorv Vyas, Angelos Katharopoulos, and François Fleuret. Fast transformers with clustered attention. In *Advances in Neural Information Processing Systems*, 2020. 2

[45] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 2

[46] Jianchao Yang, John Wright, Thomas Huang, and Yi Ma. Image super-resolution as sparse representation of raw image patches. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008. 1, 2

[47] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, 2010. 1, 2

[48] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, pages 711–730. Springer, 2010. 5

[49] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Learning a single convolutional super-resolution network for multiple degradations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3262–3271, 2018. 5, 6

[50] 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. 5, 6, 7

[51] Yulun Zhang, Kunpeng Li, Kai Li, Bineng Zhong, and Yun Fu. Residual non-local attention networks for image restoration. *arXiv preprint arXiv:1903.10082*, 2019. 1, 2, 5, 6

[52] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2472–2481, 2018. 5, 6, 7

[53] Zheng Zhang, Yong Xu, Jian Yang, Xuelong Li, and David Zhang. A survey of sparse representation: Algorithms and applications. *IEEE Access*, 3:490–530, 2015. 2