KNN Local Attention for Image Restoration

Hunsang Lee¹, Hyesong Choi², Kwanghoon Sohn¹, Dongbo Min²†
¹Yonsei University, Korea, ²Ewha W. University, Korea
hslee91@yonsei.ac.kr, hyesongchoi2010@gmail.ac.kr, khsohn@yonsei.ac.kr, dbmin@ewha.ac.kr

Abstract

Recent works attempt to integrate the non-local operation with CNNs or Transformer, achieving remarkable performance in image restoration tasks. The global similarity, however, has the problems of the lack of locality and the high computational complexity that is quadratic to an input resolution. The local attention mechanism alleviates these issues by introducing the inductive bias of the locality with convolution-like operators. However, by focusing only on adjacent positions, the local attention suffers from an insufficient receptive field for image restoration. In this paper, we propose a new attention mechanism for image restoration, called k-NN Image Transformer (KiT), that rectifies the above mentioned limitations. Specifically, the KiT groups k-nearest neighbor patches with locality sensitive hashing (LSH), and the grouped patches are aggregated into each query patch by performing a pair-wise local attention. In this way, the pair-wise operation establishes non-local connectivity while maintaining the desired properties of the local attention, i.e., inductive bias of locality and linear complexity to input resolution. The proposed method outperforms state-of-the-art restoration approaches on image denoising, deblurring and deraining benchmarks. The code will be available soon.

1. Introduction

Image restoration aims to recover a clean image from various type of degradations (e.g. noise, blur, rain, and compression artifacts), which has a huge impact on the performance of downstream tasks such as image classification [14,56], object detection [22,46], segmentation [4,10], and to name a few. It is a highly ill-posed inverse problem as there may exist multiple number of solutions for a single degraded image. Recent restoration works [17,36,76] attempt to establish a mapping relation between clean and degraded images by leveraging the representation power of the convolutional neural networks (CNNs). A series of local operations used in the CNNs is, however, inherently less capable of capturing a long-range dependency, exhibiting certain limitations in deliberating global information over an entire image. To enlarge the receptive field, increasing network depth [51], dilated convolution [66], and hierarchical architecture [40] have been proposed, but the receptive field still does not secure global information as it is limited to local regions. Recently, non-local operation, which mostly contributed to non-learning based restoration approaches [5,15], has again emerged as a promising solution with the success of non-local neural networks [58]. As similar patterns tend to repeat within a natural image, non-local self-similarity of computing the response at a single position by weighted sum of all positions has served as an important cue for an image restoration [16,28,32,37,38,43,53,77,78]. A non-local self-similarity of [58] could capture the long-range dependency within deep networks, but the quadratic complexity with respect to the input feature resolution limits the network capacity. Consequently, it is employed only in relatively low-resolution feature maps of specific layers [16,32,77].

More recently, Vision Transformer (ViT) [18] proposed a new approach to apply the global attention mechanism, which can be viewed as the non-local operation, of the Transformer [55] to vision tasks by splitting an image into...
a set of non-overlapping patches, embedding into the feature space, and feeding them into multiple transformer layers to model global self-similarities among patches (Fig. 1 (a)). ViT achieved a pleasing trade-off between accuracy and computational complexity in the image classification task, but the quadratic complexity with respect to the input feature resolution still makes it nearly infeasible to apply the transformer to dense prediction tasks. To overcome this limitation, different from ViT that maintains feature resolutions across the entire network, some approaches [45,57] proposed a hierarchical architecture to exploit multi-scale feature maps that are suitable for dense prediction tasks. However, they focus only on capturing global self-similarity, and their capability in exploring locality that is essential for image restoration is inferior to that of CNNs.

In this context, numerous methods have been proposed to introduce the inductive bias of locality into transformer architectures [30,33,59,61,63]. Among them, local attention is considered in recent works [31,33,54,59,67] at the cost of restricting the receptive field in the transformer. These approaches propose the local self-attention module, achieving a linear complexity to the input feature resolution (Fig. 1 (b)). Since they constrain the self-attention computation only within a local patch, a shifting approach [31,33,59] is additionally applied to exchange information across non-overlapping patches. However, it considers only neighboring patches and thus still has insufficient receptive field.

In this paper, we propose a novel non-local image restoration method, called \(k\)-NN Image Transformer (KiT), that successfully captures locality while explicitly establishing non-local connectivity by considering the local attention of \(k\) nearest neighbor \((k\text{-NN})\) patches. To remedy the lack of the long-range dependency inherent in the local attention, the proposed method considers \(k\) matched patches that generate non-local connectivity between patches of different positions. To be specific, the KiT first searches a set of similar patches for each base patch with \(k\text{-NN}\) matching, and then sets the base patch as query and \(k\) matched patches as key and value for applying pair-wise attention locally, as shown in Fig. 1 (c). This enables our method to apply the local attention over an entire image while maintaining a linear complexity with respect to the feature resolution. Additionally, the inductive bias of locality enhances local feature extraction capability. As shown in Fig. 2, our method consists of a series of \(k\text{-NN}\) transformer block (KTB), and adopts U-shaped hierarchical architecture for efficiently leveraging multi-scale features. Comprehensive experiments on various image restoration tasks demonstrate the effectiveness of the proposed method over state-of-the-art methods.

2. Related Works

Non-local image restoration. Non-local operation has been widely used in the image restoration. In classical approaches [15,35], a set of pixels grouped by self-similarity contributes to an output filtered response. Recently, with the success of non-local neural networks [58], some methods [16,32,77] attempted to integrate the non-local operation into CNNs for image restoration tasks by establishing the long-range dependency with the global self-attention. However, its expensive computational cost limits the spatial resolution of feature maps or network depth. To reduce computational cost, sparse connections were used in [28,37,38,43,53,78] instead of full connections within the input feature map. \(N\text{^2Net}\) [43] and GCDN [53] find \(k\)-nearest neighbors that are close in the embedding space in a learnable manner, and aggregate them for an efficient computation. DAGL [38] dynamically selects the number of neighbors for each query which has distinct distributions according to an image content. \(\text{IGNN}\) [78] and CPNet [28] finds \(k\)-NN patches among cross-scale feature maps by considering both sparseness and cross-scale patch recoverability. Nevertheless, aforementioned approaches have the quadratic complexity for \(k\)-NN matching that heavily slows the entire process. NLSN [37] reduces the complexity of \(k\)-NN matching process to be asymptotic linear by performing non-local sparse attention with locality sensitive hashing (LSH). But, as the NLSN [37] approximated full connection of the global attention in pixel-level, local information can not be captured in their attention module.

Vision Transformer. In [18], the Transformer architecture [55], originally proposed for natural language processing, was applied to the image classification task. This method, called Vision Transformer (ViT), is remarkable at capturing the long range dependencies by applying global attention to image patches, but is not suitable for dense predictions due to the quadratic complexity to an input spatial resolution. Unlike ViT that maintains a fixed spatial resolution across the entire architecture, the hierarchical architecture, where feature resolutions are progressively reduced, is adopted for conducting the dense prediction more effectively [9,45,57,62]. PVViT [57] builds pyramid feature maps with spatial reduction attention (SPA) layer. IPT [9] and DPT [45] propose an encoder-decoder architecture to recover fine-grained predictions. However, these approaches based on the global attention lacks the capability that explores locality essential for image restoration. Lately, Swin Transformer [33] leverages the local attention with a shifting approach for patch connection and achieves a competitive performance on object detection and segmentation with low complexity. As the local attention module generates attention weights among adjacent elements only, the computational complexity is linear to the spatial resolution and the inductive bias of locality is injected into the attention. Uformer [59] and SwinIR [31] adopt the local attention for image restoration tasks, demonstrating impressive results. However, the shifting approach still has a limited recep-
The local attention mechanism [31, 33, 54, 59, 67] reduces the complexity by computing attention within a local patch. An input feature map $X$ is split into non-overlapping patches, satisfying $X = \{x_i \in \mathbb{R}^{r \times C} | i = 0, \ldots, N - 1\}$. The local attention is computed within each patch individually

$$o_i = \text{softmax}\left(\frac{\phi(x_i)\theta(x_i)^T}{\sqrt{C}}\right)\psi(x_i), \quad (2)$$

where $o_i$ is an output patch corresponding to $x_i$. Note that the learnable projection functions $\phi, \theta$ and $\psi$ project $r^2$ elements with a size of $C$, unlike ViT projecting $N$ elements with a size of $r^2C$, and are shared for all patches. The local attention achieves the linear complexity $O(r^4NC)$ to the input feature resolution. However, as Eq. (2) is applied to each patch separately, no information is exchanged across patches. Thus, a shifting approach [31, 33, 59] is sequentially applied for imposing patch connectivity among neighbor patches with an enlarged receptive field. Nevertheless, as only neighbor patches contribute to the query patch, the receptive field is still limited.

We overcome this limitation by leveraging $k$-NN in the computation of the local attention, termed $k$-NN local attention. In order to impose the non-local connectivity in computing the local attention, we utilize $k$-NN search to seek a set of patch candidates used for computing the local attention. By conducting the pair-wise local attention between a query patch and $k$ matched patches, the proposed method captures locality efficiently while establishing non-local connectivity essential for image restoration.

3.2. Overall Pipeline

The overall framework for image restoration is shown in Fig. 2. To restore a degraded image, we first conduct three convolutions to a degraded input image $I_t$, and then pass it through three stages of the encoder network and the decoder.
network. Each stage is comprised of the patch partition, k-NN Transformer Blocks (KTB), and an interpolation layer. The patch partition operation splits the input feature map $X$ into non-overlapping patches with the patch size $r$, satisfying $X = \{ x_r \in \mathbb{R}^{r \times r C} \mid i = 0, \ldots, N - 1 \}$.  

In the KTB, the split patches are normalized and fed to $k$-NN local attention (KLA) for non-local aggregation. The KLA first establishes $k$ patches based on the fact that patches with similar patterns frequently appear within an image and aggregating them is beneficial to the image restoration. Note that the existing sparse attention approaches [26, 37] cluster pixels into separate groups to approximate global attention, and thus naturally the lack of the locality, which is a drawback of the global attention, still remains in their attention module. On the other hand, the proposed method computes the sparse similarity between query patch and $k$ patches with the pair-wise local attention, and the inductive bias of the locality is reflected in the proposed attention module.  

The proposed network has a U-shaped hierarchical architecture for taking patterns of various scales into account. The aggregated features pass through an interpolation layer (downsampling for encoder and upsampling for decoder). In each stage of the decoder, input feature maps are concatenated with corresponding encoder features for recovering fine details. At the end of the network, three convolutions are conducted to predict a restored image from the output feature map.

### 3.3. KTB: $k$-NN Transformer Block

A layer normalization (LN) is applied to each patch, and then the $k$-NN local attention (KLA) conducts local attention with query patch and matched patches as key and value. To enhance locality of the network, depth-wise convolution (DW) [13] is employed together with multi-layer perceptron (MLP) in the feed-forward network (FFN) [30], as depicted in Fig. 2 (c). Formally, the KTB is written as:

$$\hat{X}^l = \text{KLA}(\text{LN}(X^{l-1})) + X^{l-1},$$

(3)

$$X^l = \text{FFN}(\text{LN}(\hat{X}^l)) + \hat{X}^l,$$

(4)

where $\text{FFN}(X) = \text{MLP}(\text{DW}(\text{MLP}(X)))$. In $l$-th block of each stage ($l = 0, \ldots, b - 1$), the output feature map $X^{l-1}$ from the previous block is normalized and fed into KLA. The intermediate feature $\hat{X}^l$ is computed via a non-local aggregation of $k$ similar patch features and residual connection. The bottleneck stage is identical to the KTB, except that the interpolation layer is not used and $k$ is set to 1.

#### $k$-nearest neighbor matching

A brute-force $k$-NN matching requires computing a pair-wise distance between two patches. As this pair-wise distance involves the quadratic complexity to an input length, we leverage the locality sensitive hashing (LSH) [2] that has linear computational complexity. The LSH projects split patches into an unit hypersphere to establish buckets. Assuming there are $m$ hash buckets, a hash value $L(x)$ is assigned by multiplying random rotation matrix $R \in \mathbb{R}^{N \times m/2}$ to a spherically projected patch $x$ as:

$$L(x) = \arg \max ([xR; -xR]),$$

(5)

where $[::]$ indicates the concatenation of two elements. With this hashing operation, patches with high correlation are very likely to receive the same hash value (in the same hash bucket), and vice versa. However, as LSH depends on random rotation matrix, similar patches may occasionally fall in different hash buckets. To cope with this issue, multi-round LSH is adopted where LSH is applied with different random rotation matrix $h$ times.

#### KLA: k-NN local attention

As shown in Fig. 3, similar patches are grouped according to the assigned hash values. To make only patches with the same hash value contribute...
to a query patch efficiently, we first sort patches according to hash values, and then partition the sorted patches into chunks each involving $k$ patches (equal to the number of NN patches) for batching purpose, so that the local attention is performed on only patches in the same chunk. We denote $\pi : n \to n$ be a permutation that sorts the patches in ascending order of hash values:

$$\pi(x_p) < \pi(x_q) \Rightarrow L(x_p) \leq L(x_q). \quad (6)$$

For the sake of a simplicity, we define $\tilde{x}$ as a sorted patch where $\tilde{x}_p$ is equal to $x_{\pi(p)}$. Then, $i$-th chunk $\mathcal{P}_i$ for $i = 0, \ldots, N/k$ contains $k$ patches,

$$\mathcal{P}_i = \{\tilde{x}_{ki}, \tilde{x}_{ki+1}, \tilde{x}_{ki+2}, \ldots, \tilde{x}_{ki+k-1}\}. \quad (7)$$

The local attention is then conducted within patch pairs in the chunk for non-local aggregation. Sorted input patches $X$ are projected into query, key and value with the learnable projection functions $\phi, \theta$ and $\psi : \mathbb{R}^{r \times c} \to \mathbb{R}^{r \times c}$, respectively. As there are $k$ patches in a chunk, the local attention is conducted $k^2$ times. The pair-wise local attention output for $p$-th patch as a query where $q$-th patch is used as key and value is defined as:

$$o_{p,q} = \text{softmax} \left( \frac{\phi(\tilde{x}_p)\theta(\tilde{x}_q)^T}{\sqrt{C}} \psi(\tilde{x}_q) \right). \quad (8)$$

For instance, denoting $\phi(\tilde{x}_1)$ as a query patch in chunk $\mathcal{P}_0$ of Fig. 3, there are $k$ output patches $\{o_{1,j}|j = 0, \ldots, k-1\}$. Different from ViT [18] that performs self-attention of all patches, we perform the local attention, but $k$ times and in the pair-wise manner that computes affinity matrix between two patches (query and key) for enhancing locality as described in Fig. 1 (c).

As $k$ output patches for a query patch are computed, the pair-wise outputs should be aggregated into the query patch. The output patch $o_p$ for the input patch $\tilde{x}_p$ is computed by weighted sum as:

$$o_p = \sum_{j \in \mathcal{N}_p} w_{p,j} \cdot o_{p,j}, \quad (9)$$

where $w_{p,j}$ is a pair-wise relative similarity between patches, and $\mathcal{N}_p$ is a set of patch indices of a chunk to which the query patch $\tilde{x}_p$ belongs,

$$w_{p,q} = \frac{\phi(\tilde{x}_p) \cdot \theta(\tilde{x}_q)}{\sum_{j \in \mathcal{N}_p} \phi(\tilde{x}_p) \cdot \theta(\tilde{x}_j)}. \quad (10)$$

Here, $\tilde{x}_p \in \mathbb{R}^{r \times c}$ represents the flatten patch of $\tilde{x}_p$. As the number of patches in a hash bucket is often indivisible by chunk size in practice, the patches with the same hash value may fall into nearby chunks. To deal with it, similar to [26], we allow the previous chunks to contribute to the current chunk containing the query patch, e.g. $\mathcal{P}_{i-1}$ for $\mathcal{P}_i$. Thus, the local attention is conducted $2k$ times for each query patch.

### 3.4. Training Loss

Following existing image restoration approaches [32, 76], the proposed network also predicts a residual image $I_r$ from the degraded input image $I_d$. The objective is to recover clean image $I$ satisfying $I = I_d + I_r$. We leverage Charbonnier loss [8] $L_{\text{char}}$ and an edge loss $L_{\text{edge}}$ for optimizing the network,

$$L_{\text{char}} = \sqrt{\|I - (I_d + I_r)\|^2 + \epsilon^2},$$

$$L_{\text{edge}} = \sqrt{\|\nabla I - \nabla(I_d + I_r)\|^2 + \epsilon^2},$$

$$L = L_{\text{char}} + \lambda L_{\text{edge}}.$$  

where $\epsilon$ is empirically set to $10^{-3}$ for all experiments and $\nabla$ represents the Laplacian function. The total loss $L$ is defined with $L_{\text{char}}$ and $L_{\text{edge}}$, where a hyper-parameter $\lambda$ controls the ratio of the two losses.

### 4. Experiments

#### 4.1. Implementation Details

The proposed KiT was implemented in PyTorch. We trained the whole networks on the batches of 16 images cropped to $128 \times 128$ for 300 epochs using AdamW optimizer [34]. The learning rate was set to $1 \times 10^{-4}$ initially, and the linear warm-up strategy and cosine annealing for decreasing the learning rate were adopted. The chunk size $k$ (equal to the number of NN patches) and patch size $r$ were set to 4 by default. In the bottleneck stage, $k$ is set to 1 since there are only a few patches (e.g. the number of patches is $4 \times 4$ when $HW$ is $256 \times 256$). The number of KTB in each stage, $b$, was set to 2 in all stages. In the KLA, the number of hashes, $h$, was set to 4 for multi-round LSH. We validated the performance of the proposed method on various image restoration tasks such as image denoising, deblurring and deraining. For the performance evaluation, the PSNR and SSIM were measured on the RGB space for denoising and deblurring. In deraining, the evaluation was done on the Y channel of the YCbCr color space, following previous works [24, 72]. More results are provided in the supplementary materials.

#### 4.2. Image Denoising

We trained the KiT with the SIDD [1] dataset containing 320 high-resolution images with realistic noise. Tab. 1 shows the quantitative evaluation of real noise removal on the SIDD [1] and DND [42] datasets. The evaluation results include the classical denoising method [15], CNN-based methods [3, 6, 11, 23, 25, 68, 70–72, 76], self-attention based methods [38] and transformer-based methods [59]. As DND [42] dataset does not provide ground-truth labels, the results were obtained from official benchmark. The proposed method outperforms the state-of-the-art methods both
on SIDD datasets and achieves competitive performance on the DND dataset. As the DND dataset does not provide any training data, the performance in Tab. 1 is achieved using the network trained on SIDD dataset, proving the robustness of the proposed method. Fig. 4 shows the denoised images with various state-of-the-art methods. While existing methods output restored images with loss of details, the proposed method successfully restores degraded images with fine-detailed structures thanks to the capability of capturing locality with non-local connectivity.

4.3. Image Deblurring

Tab. 2 reports the image deblurring performance on the GoPro dataset [39]. The GoPro dataset provides synthetic blurry images where each image is obtained by averaging successive sharp images. For training, 2,103 images of the GoPro [39] dataset were used, and 1,111 images of the GoPro [39], 3,758 images of RealBlur [48] and 2,025 images of the HIDE [49] datasets were evaluated. The outstanding performance on the PSNR and SSIM metrics validated that the proposed method is also beneficial to restoring blurry images. Fig. 5 shows restored images from blur artifacts in the GoPro [39] dataset. It is easily found that our results capture sharp and fine details whereas other methods are unable to deal with high-frequency details.

4.4. Image Deraining

Following the experimental setup of [24], 13,712 clean-rain image pairs sampled from multiple datasets [20, 29, 64, 74, 75] were used to train the network for image deraining. We evaluated the deraining results on five datasets, Test100 [75], Rain100H [64], Rain100L [64], Test2800 [20], and Test1200 [74]. While SPAIR [44] that leverages extra distortion-guided networks shows competitive results with the proposed method in terms of PSNR, the proposed KIT achieves a higher SSIM, demonstrating that fine-details can be better restored. As shown in the results of Fig. 4 and Fig. 6, the proposed method has an advantage of dealing with repeated textures thanks to the KLA based aggregation. In rainy images, as many patches with similar patterns exist in the image, the proposed method shows out-
4.5. Ablation Study

We conducted the ablation studies to analyze the performance of our method at various aspects. All experiments were conducted on SIDD [1] for image denoising task.

**Computational complexity.** The proposed method is comprised of multi-round hashing for $k$-NN search, feature projection, and pair-wise local attention. The multi-round hashing is performed to the input feature patches by multiplying random rotation matrix $R$, which has $O(hNCm)$ complexity. Then, each patch is projected to the query, key, and value with the learnable projection functions $\phi$, $\theta$ and $\psi : \mathbb{R}^{r^2 \times C} \rightarrow \mathbb{R}^{r^2 \times C}$, whose complexity is $O(NC^2)$. The complexity of computing the local attention between all patch pairs is $O(khr^4N)$. Thus, all operations takes linear computation with respect to the input feature resolution.

**Visualization of the $k$-NN patches.** Our method aims to preserve fine details while achieving non-local connectivity efficiently, achieved by aggregating patches with similar characteristics. To visually validate this, we further visualize the patches used for KLA in Fig. 7. The left most images are divided into non-overlapping patches, where the patches marked with color boxes represent query patches for visualization. As the KLA leverages LSH for $k$-NN search, similar $k$ patches are grouped with a chunk in the right figures. The $k$ patches belonging to the same chunk are also marked with the same color boxes. With red and green boxes have similar patterns, while the patches with blue boxes include non-textured areas, proving that the LSH finds visually similar patches effectively.
Table 3. The quantitative results of image deraining. The widely used five datasets [20, 64, 74, 75] are used for evaluation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
<td>SSIM</td>
<td>PSNR</td>
<td>SSIM</td>
<td>PSNR</td>
<td>SSIM</td>
</tr>
<tr>
<td>DerainNet [19]</td>
<td>22.77 0.810</td>
<td>14.92 0.592</td>
<td>27.03 0.884</td>
<td>24.31 0.861</td>
<td>23.38 0.835</td>
<td>22.48 0.796</td>
</tr>
<tr>
<td>SEMI [60]</td>
<td>22.35 0.788</td>
<td>16.56 0.486</td>
<td>25.03 0.842</td>
<td>24.43 0.782</td>
<td>26.05 0.822</td>
<td>22.88 0.744</td>
</tr>
<tr>
<td>DIDMDN [74]</td>
<td>22.56 0.818</td>
<td>17.35 0.524</td>
<td>25.23 0.741</td>
<td>28.13 0.867</td>
<td>29.65 0.901</td>
<td>24.58 0.770</td>
</tr>
<tr>
<td>UMLR [65]</td>
<td>24.41 0.829</td>
<td>26.01 0.832</td>
<td>29.18 0.923</td>
<td>29.97 0.905</td>
<td>30.55 0.910</td>
<td>28.02 0.880</td>
</tr>
<tr>
<td>RESCAN [27]</td>
<td>25.00 0.835</td>
<td>26.36 0.786</td>
<td>29.80 0.881</td>
<td>31.29 0.904</td>
<td>30.51 0.882</td>
<td>28.59 0.857</td>
</tr>
<tr>
<td>PreNet [47]</td>
<td>24.81 0.851</td>
<td>26.77 0.858</td>
<td>32.44 0.950</td>
<td>31.75 0.916</td>
<td>31.36 0.911</td>
<td>29.42 0.897</td>
</tr>
<tr>
<td>MSPFN [24]</td>
<td>27.50 0.876</td>
<td>28.66 0.860</td>
<td>32.40 0.933</td>
<td>33.64 0.938</td>
<td>32.89 0.916</td>
<td>32.73 0.921</td>
</tr>
<tr>
<td>DIDMDN [72]</td>
<td>30.27 0.897</td>
<td>30.41 0.890</td>
<td>36.40 0.965</td>
<td>33.34 0.936</td>
<td>33.04 0.922</td>
<td>32.91 0.926</td>
</tr>
<tr>
<td>UMRL [65]</td>
<td>30.35 0.909</td>
<td>30.95 0.892</td>
<td>36.93 0.969</td>
<td>33.34 0.936</td>
<td>33.04 0.922</td>
<td>32.91 0.926</td>
</tr>
</tbody>
</table>

Table 4. Ablation study of the number of patches $k$ and hash rounds $h$.

<table>
<thead>
<tr>
<th>PSNR</th>
<th>$h$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>38.79</td>
</tr>
<tr>
<td>2</td>
<td>39.58</td>
</tr>
<tr>
<td>4</td>
<td>39.69</td>
</tr>
<tr>
<td>8</td>
<td>39.74</td>
</tr>
<tr>
<td>16</td>
<td>39.75</td>
</tr>
</tbody>
</table>

Table 5. Ablation study of the shared projection $\phi = \theta$ (KIT$_S$).

<table>
<thead>
<tr>
<th>Method</th>
<th>SIDD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
</tr>
<tr>
<td>KIT</td>
<td>39.80</td>
</tr>
<tr>
<td>KIT$_S$</td>
<td>39.75</td>
</tr>
</tbody>
</table>

The number of $k$ and $h$. The chunk size $k$ determines the maximum number of patches used for performing the local attention with the query patch. The $h$ hash rounds are used to reduce the probability that similar patches fall into different hash buckets. As the two hyper-parameters do not affect the number of network parameters and are only related to the memory and computational complexity, the network capacity can be flexibly adjusted according to computational resources. Tab. 4 shows the denoising performance of the proposed method according to the two hyper-parameters. The best performance was achieved when the two hyper-parameters are set to 16, but, we set $k$ and $h$ to 4 as it has comparable performance with relatively low computation.

Sharing query and key. In the existing methods [26, 37] that leverage LSH for sparse global attention, the projection functions for query and key should be shared, i.e., $\phi = \theta$, and thus the shared attention masks out the query as the dot-product of a query with itself almost always overwhelms the dot product of a query with a key at other positions. Contrarily, our method has no such a constraint for sharing the projection functions. For the purpose of ablation study, we conducted additional experiments of sharing the projection functions for query and key in Tab. 5, denoted as KIT$_S$. When using the shared projection, the overall performance was slightly reduced.

5. Conclusion

In this paper, we have presented a novel non-local image restoration method. Specifically, the $k$-NN local attention (KLA) conducts the pair-wise local attention among similar patches with $k$-NN matching. The KLA holds the inductive bias of locality while establishing the non-local connectivity with the linear computational complexity to the input spatial resolution. The proposed method outperforms the state-of-the-art methods on various image tasks, in terms of quantitative/qualitative performance. As the number of NN patches and hash rounds that determine the network capacity are independent of the network parameters, the flexible adjustment of the network capacity is feasible.

Limitations. The proposed method only considers the pair-wise local attention between patches of the same scale. The cross-scale attention can be an interesting methodology that further improves the restoration performance. We will continue to investigate network that integrates in cross-scale attention for our cross-position based model.
References


[20] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley. Removing rain from single images via a deep detail network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3855–3863, 2017. 6, 8


[25] Yoonsik Kim, Jae Woong Soh, Gu Yong Park, and Nam Ik Cho. Transfer learning from synthetic to real-noise denoising with adaptive instance normalization. In Proceedings of


