1. Introduction

The goal of single image super-resolution (SR) is to reconstruct high-resolution (HR) images from low-resolution (LR) images. Many deep learning-based methods have worked in this field. In particular, several image restoration studies [16, 27, 55, 61, 63, 66] have adapted the window self-attention (WSA) proposed by Swin Transformer (Swin) [12] as it integrates long-range dependency of Vision Transformer [13] and locality of conventional convolution. However, two critical problems remain in these works. First, the receptive field of the plain WSA is limited within a small local window [52, 56, 58]. It prevents the models from utilizing the texture or pattern of neighbor windows to recover degraded pixels, producing the distorted images. Second, recent state-of-the-art SR [9, 27, 61, 66] and lightweight SR [6, 15, 35, 63, 66] networks require intensive computations. Reducing operations is essential for real-world applications if the parameters are kept around a certain level (e.g., 1M, 4MB sizes), because the primary consumption of semiconductor energy (concerning time) for neural networks is related to Multi-Adds operations [17, 47].

To overcome these problems, we define the N-Gram context as the interaction of neighbor local windows. Neighbor uni-Gram embeddings interact with each other by sliding-WSA to produce the N-Gram context features before window partitioning. The uni-Gram embeddings result from a channel-reducing group convolution [10] to decrease the complexity of N-Gram interaction (see Fig. 3c). Our N-Gram context efficiently expands the receptive field of WSA for recovery tasks. This work introduces N-Gram to low-level vision with Transformers for the first time, inspired by the following facts: N-Gram language models treat the text statistically [3]. Since images have heavy spatial redundancy, some degraded pixels can be recovered from contextual information of neighbor pixels [18].

As shown in Fig. 1, our work progresses in two tracks. Mainly, to solve the problem of the intensive operations in SR, we propose an efficient N-Gram Swin Transformer (NGswin). As illustrated in Fig. 3a, NGswin consists of five components: a shallow module, three hierarchical encoder stages (with patch-merging) that contain NSTBs (N-Gram Swin Transformer Blocks), SCDP Bottleneck taking multi-scale outputs of the hierarchical encoder. Experimental results show that NGswin outperforms previous leading SR methods with an efficient structure. (Right) Our proposed N-Gram context improves different Swin Transformer-based SR models.
tlenck (pixel-Shuffle, Concatenation, Depth-wise convolution, Point-wise projection), a small decoder stage with NSTBs, and a reconstruction module. NSTBs employ our N-Gram context and the scaled-cosine attention proposed by Swin V2 [31]. SCDP bottleneck, which takes multiscale outputs of the encoder, is a variant of bottleneck from U-Net [46]. Experimental results demonstrate that the components above contribute to the efficient and competitive performance of NGswin. Secondly, focusing on improved performances, we apply the N-Gram context to other Swin-based SR models, such as SwinIR-light [27] and HNCT [16]. Notably, SwinIR-NG (improved SwinIR-light with N-Gram) establishes state-of-the-art lightweight SR.

The main contributions of this paper are summarized as:

1. We introduce the N-Gram context to the low-level vision with Transformer for the first time. It enables the SR networks to expand the receptive field to recover each degraded pixel by sliding-WSA. For efficient calculation of N-Gram WSA, we produce uni-Gram embeddings by a channel-reducing group convolution.

2. We propose an efficient SR network, NGswin. It exploits the hierarchical encoder (with patch-merging), an asymmetrically small decoder, and SCDP bottleneck. These elements are critical for competitive performance in the efficient SR on ×2, ×3, and ×4 tasks.

3. The N-Gram context improves other Swin Transformer methods. The improved SwinIR-NG achieves state-of-the-art results on lightweight SR.

2. Related Work

Efficient SR. Many single image super-resolution (SR) studies have increased network efficiency. CARN [2] introduced cascading residual blocks. IMDN [23] used information multi-distillation and selective feature fusion. LatticeNet [37] utilized the lattice filter that varies Fast Fourier Transformation. ESRT [35] combined convolutional neural networks (CNN) and channel-reducing Transformers [53]. SwinIR-light [27] and HNCT [16] appended CNN to Swin Transformer [32]. SRPN-Lite [65] applied the network pruning technique [45] on EDSR-baseline [38], a CNN-based lightweight SR model. Most recently, ELAN-light [53] utilized group-wise multi-scale self-attention.

N-Gram. In language model (LM), N-Gram is a sequence of consecutive characters or words. The size N is typically set to 2 or 3 [39]. The N-Gram LM that considers a longer span of context in sentences was operating well statistically in the past. Even some deep learning LMs still adopted N-Gram. Sent2Vec [43] used N-Gram embeddings to learn sentence embedding by averaging word-embedding. To learn the sentence representation better, [33] computed the word N-Gram context by recurrent neural networks (RNN) and passed it to the attention layer. ZEN [12,49] trained a BERT-styled [11] N-Gram encoder for all possible N-Gram pairs from the Chinese or Arabic lexicon to convey salient pairs to the character encoder. Meanwhile, some high-level vision studies also adopted this concept. The Pixel N-grams approach [26] saw N-Gram in pixel level and a single (horizontal or vertical) direction. View N-gram Network [20] regarded consecutive (along time steps) multi-view images of a 3D object as an N-Gram. In contrast, our N-Gram considers bi-directional 2D information in local window level, given a single image for low-level vision.

Swin Transformer. Swin Transformer [32] (SwinV1) proposed window self-attention (WSA) that computes self-attention within non-overlapping local windows, to avoid quadratic time-complexity to the resolution of feature map. SwinV1 also placed a shifted window scheme in consecutive layers, capturing interaction across windows. Some studies [16,27] utilized effective SwinV1 for SR. The revised version [31] (SwinV2) modified SwinV1. For advanced model capacity with milder optimization, SwinV2 introduced residual post-normalization and scaled-cosine attention (in Eq. (2), (3)) instead of pre-normalization configuration and scaled-dot-product attention.

3. Methodology

3.1. Problem Verification

The plain window self-attention (WSA) suffers from limited receptive field, as criticized in many recent studies [13,52,56,58,61,62,66]. We observe this issue by visualizing feature maps after self-attention. The similar pixel values from self-attention of the deeper layer tend to recover into a homogeneous pattern or texture. In (h) of Fig. 2, however, the patterns in the red box and its neighbors differ (problem α), causing the distortions in (e). Problem α stems from problem β: The plain WSA (i.e., w/o N-Gram) of the shallow layer is limited to only a local window. It cannot utilize surrounding patterns to infer the recovery pattern of each window. In (f) and (g), the distinctive colors across

![Figure 2](image_url). The feature maps after window self-attention in each intermediary layer of NGswin with and without N-Gram.
Given a low-resolution (LR) image \( R \) stand for height, \( H \) × \( W \), to 2 indicates a bi-Gram that combines a low-resolution (LR) image. In this paper, "resolution" indicates height and width of feature maps, \( H, W \), and \( M \) are kernel size, stride, padding, and local window size, respectively. The dimensionality reduction through uni-Gram embedding makes \textit{sliding}-WSA efficient. The bi-directional contexts share \textit{sliding}-WSA weights. For window-wise sum, a value in \( z_{\text{WSA}} \) is equally added to \( M^2 \) pixels in one local window at the same position.

Figure 3. Overall architecture of NGswin and NSTB (N-Gram Swin Transformer Block). (a) We adopt an asymmetric U-Net architecture. SCDP Bottleneck (pixel-Shuffle, Concatenation, Depth-wise convolution, and Point-wise projection), a variant of the U-Net bottleneck, takes multi-scale outputs of encoder stages, including the shallow module. (b) Our proposed N-Gram method is implemented in NSTB.

We also employ scaled-cosine attention and post-normalization. \( k, s, p, \) and \( M \) are kernel size, stride, padding, and local window size, respectively. The dimensionality reduction through uni-Gram embedding makes \textit{sliding}-WSA efficient. The bi-directional contexts share \textit{sliding}-WSA weights. For window-wise sum, a value in \( z_{\text{WSA}} \) is equally added to \( M^2 \) pixels in one local window at the same position.

We also employ scaled-cosine attention and post-normalization. (a) N-Gram in text. (b) N-Gram in image (proposed).

Figure 4. N-Gram in text and image \((N = 2)\). (a) The underlined words are the target words and the non-underlined words are neighbors of the target words. (b) Each local window is defined as uni-Gram. The lower-right (or upper-left) local windows are defined as forward (or backward) N-Gram neighbors.

The adjacent windows reveal \textit{problem} \( \beta \). This seems to be resolved in the deeper layers, but it fails to overcome \textit{problem} \( \alpha \). To address this, we propose the N-Gram context to compensate this vulnerability. Our N-Gram attention can consider broad regions (\textit{i.e.}, surrounding patterns) beyond a window. In (a)-(d), the semantically relevant areas yield similar attention results, producing more accurate details. This crucial advantage solves both \textit{problem} \( \alpha \) and \( \beta \).

3.2. Definition of N-Gram in Image

N-Gram in text. As shown in Fig. 4a, the N-Gram language model views the consecutive forward, backward, or bi-directional words as the N-Gram of the target word. The words are independent of each other for uni-Gram (\textit{i.e.}, word-embedding), but they interact with each other by averaging word-embeddings \([43]\), RNN \([33]\), or attention \([12]\) when considering N-Gram. In contrast, an N-Gram composed of a particular word pair (\textit{e.g.}, "office work") never interacts with the other N-Gram combinations when producing a N-Gram feature.

N-Gram in image. N-Gram in an image should have the properties above. Accordingly, we define a uni-Gram as a non-overlapping local window in Swin Transformer, within which the pixels interact with each other by self-attention (SA). N-Gram is defined as the larger window, including neighbors of each uni-Gram. To sum up, pixels of each uni-Gram and uni-Grams of each N-Gram in image correspond to characters of each word and words of each N-Gram in text, respectively. As depicted in Fig. 4b, setting N-Gram size \( N \) to 2 indicates a bi-Gram that combines a local window (green area) and its neighbor windows (red areas) at lower-right (forward) or upper-left (backward). The N-Gram interactions will be explained in Sec. 3.4.

3.3. Overall Architecture of NGswin

As illustrated in Fig. 3a, we adopt U-Net \([46]\) architecture: hierarchical encoder stages, a bottleneck layer, a decoder stage, and skip-connection from the encoder to the decoder at the same resolution\(^1\). However, our network’s encoder and decoder are asymmetric, which indicates a significantly smaller decoder \([18, 44]\).

Encoder. Given a low-resolution (LR) image \( I_{LR} \in \mathbb{R}^{3 \times H \times W} \), a shallow module (a \( 3 \times 3 \) convolution) extracts \( z_s \in \mathbb{R}^{HW \times D} \), where \( H, W, \) and \( D \) stand for height, width, and network dimension (channels), respectively. \( z_s \) is passed through three encoder stages, each composed of \( K_i \) N-Gram Swin Transformer Blocks (NSTB, Sec. 3.4)

\(^1\)In this paper, “resolution” indicates height and width of feature maps, excluding network dimension (channel).
and a $2 \times 2$ patch-merging except the last stage. We set \{\(K_1, K_2, K_3\)\} to \{6, 4, 4\} by default. The mapping function \(F^k_{enc}\) of \(k\)-th (\(1 \leq k \leq K_1\)) encoder stage is formulated as:

\[
z^k_{enc} = F^k_{enc}(z^{k-1}_{enc}), \quad z^k_{enc} \in \mathbb{R}^{HW/(2^{k-1})^2 \times D},
\]

where \(z^0_{enc}\) equals \(z_{enc_{-1}}\), which results from downsampling \(z_{enc_{-1}}\) (\(z_{enc_0} = z_s\)). In other words, the first NSTB in the 2nd or 3rd stage takes the output of patch-merging in the previous stage as input. The patch-merging follows Swin Transformer [31, 32], except that the network dimension is decreased from \(4D\) to \(D\) instead of \(2D\). Since patch-merging halves the resolutions, NGSwin consumes much fewer attention computations than state-of-the-art attention-based lightweight SR methods, as revealed in Tab. 1.

### Pooling Cascading

Following the global cascading in CARN [2], we employ a cascading mechanism (○ marks and the dashed lines in Fig. 3a) across the stages, including the shallow module. Unlike CARN, we place \(2 \times 2\) max-poolings before concatenating the intermediary features because the first and second stages halve the resolutions of features. This dense connectivity [21] reflects the flow of the information and gradient in the previous layers, which helps the network to learn meaningful representations.

### Bottleneck

All outputs from the shallow module and the last NSTBs of each encoder stage are taken by SCDP bottleneck. The bottleneck layer maps them into \(z_{scdp} \in \mathbb{R}^{HW \times D}\). A detailed explanation is in Sec. 3.5.

### Decoder

\(z_{scdp}\) is fed into a single decoder stage, which is asymmetrical smaller than the encoder [18, 44]. This means fewer stages and NSTBs in our decoder, which highly enhances the efficiency as shown in Fig. 5. It contains \(K_{dec}\) (by default, 6) NSTBs and a final layer-norm (LN) [5] that allows stable learning. The decoder NSTB architecture is the same as the encoder NSTB. As done in U-Net [40], the input to the decoder is residually connected [19] with \(z_{enc}\) of the first encoder stage. \(z_s\) and decoder output \(z_{dec} \in \mathbb{R}^{HW \times D}\) are added with a global skip-connection [3, 24, 27]. This boosts optimization and allows the reconstruction module to utilize both locality and long-range dependency.

### Reconstruction

Following [2, 24, 27, 28], the reconstruction module contains a convolution that adjusts dimension

\[2\text{We correct Mult-Adds [27] underestimated on a 1024} \times \text{720 HR image.}\]
scaled-cosine self-attention and \( N \times N \) average-pooling are computed. But the scaled-dot-product attention is used for SwinIR-light and HNCT, following their own methods. We use seq-refl-win-pad instead of trivial zero padding for \((N-1)\) size of paddings, as explained in the supplementary Sec. A.1. Subsequently, we can obtain the backward N-Gram feature \( z^b_{ng} \) by reversed seq-refl-win-pad (i.e., upper-left side padding). The computations of bi-directional N-Gram features share the sliding-WSA weights. Since the image is two-dimensional data, our N-Gram can be seen from max quad-directions (lower-right, lower-left, upper-right, and upper-left), unlike text that can be seen from max bi-directions. However, Tab. 5 demonstrates that the trade-off between performance and efficiency is optimized at the bi-directions.

Third, after the concatenation of \( z^f_{ng} \) and \( z^b_{ng} \), a \( 1 \times 1 \) convolution merges it to produce the N-Gram context \( z_{ng} \).

Finally, \( z_{ng} \in \mathbb{R}^{D \times w_h \times w_w} \) is added window-wise to the partitioned windows (size: \( M^2 \times D \times w_h \times w_w \)) from \( z^{k-1}_{enc} \). In Fig. 3c, one value in \( z_{ng} \) is equally added to \( M^2 \) pixels in one local window at the same position (marked as the same character) — i.e., the average correlations from self-attention within each N-Gram serve as bias terms of each pixel. After the four steps, NSTB follows the sequence in Fig. 3b. The window-shifts are operated in the even numbered blocks, same as in Swin Transformer.

NSTBs and patch-merging within a stage are residu-ally connected [19] from a previous layer to the next (the rounded arrows of Fig. 3a), rather than dense connections [21]. The connection to a patch-merging, however, is excepted from the third encoder stage and the decoder stage. For more, see the supplementary Sec. A.2.

3.5. SCDP Bottleneck

Many SR models [27, 42, 64] commonly never used the hierarchical encoder, which downsamples the resolutions of features after one stage. As demonstrated in Tab. 6, the hierarchical networks are inferior to the less (or non) hierarchical architectures. However, our encoder is constructed hierarchically by patch-merging. Thus, only passing the output \( z^f_{enc} \) of the last NSTB in the final encoder stage to the bottleneck makes the recovery task more challenging. The use of SCDP bottleneck, though, can convey rich representations of multi-scale features to the decoder and maintain the efficiency of NGswin. Algorithm 1 provides the pseudo-code of SCDP pipeline. It contains dimensionality rearrangements, non-linear activation functions, and LN omitted in the main text below.

SCDP stands for pixel-Shuffle, Concatenation, Depth-wise convolution, and Point-wise projection. In contrast with the bottleneck of standard U-Net that takes the output of the last encoder layer [46, 55, 59], SCDP bottleneck takes multi-scale outputs from the encoder. First, as depicted in the blue-edged triangles in Fig. 3a, for obtaining \( z^f_{enc} \) the pixel-shuffle [48] layers upsample the outputs \( z^f_{enc} \) of the last NSTB in each encoder stage into the resolution of I_{LR}. Before upsizing, \( z_s \) is iteratively max-pooled into the resolution of each \( z^{k}_{enc} \), then added to \( z^{k}_{enc} \). This process gives multi-scale information to the bottleneck. Second, all \( z^{k}_{enc} \) are concatenated in channel (network dimension) space. Third, the output passes through a \( 3 \times 3 \) depth-wise convolutional layer for learning spatial representations in each channel space. Finally, a point-wise linear projection is applied to match dimension \( D \). As a result, we get \( z_{scdp} \) and then add it to \( z^f_{enc} \) to pass it to the decoder.

4. Experiments

4.1. Experimental Setup

Training. We used 800 HR-LR (high- and low-resolution) image pairs from DIV2K [1] dataset. LR images were randomly cropped into \( 64 \times 64 \) size patches augmented by random horizontal flip and rotation (90°, 180°, 270°), as in the recent works [6, 16, 27, 63]. We minimized \( L_1 \) pixel-loss between \( I_{SR} \) and the ground truth \( I_{HR} \): \( \mathcal{L} = \| I_{HR} - I_{SR} \|_1 \), with Adam [25] or AdamW [34] optimizer. NGswin, SwinIR-NG, and HNCT-NG (improved with N-Gram) were...
Table 2. Comparison of efficient super-resolution results. D2K stands for the DIV2K dataset we used to train NGswin. DF2K indicates a merged dataset of D2K and Flickr2K [5] containing 800 + 2,650 HR-LR image pairs. 291 images dataset is from [1,5]. Multi-Adds is evaluated on a 1280 × 720 HR image. The best, second best, and third best performances are in red, blue, and underline.

Table 3. Comparison of state-of-the-art lightweight super-resolution results. SwinIR-NG is SwinIR-light improved with the N-Gram. The mark ↓ and ▼ indicates reduced-channel and DF2K, respectively. ‘Year’ indicates the publication year of each paper. The best and second best results are in red and blue.

trained by the same strategies except warm-start [29]. While we trained NGswin and SwinIR-NG from scratch on ×2 and by warm-start (using pre-trained ×2 weights) on ×3 and ×4, HNCT-NG was trained from scratch on all tasks. Other details are in the supplementary Sec. B.

Evaluation. We evaluated the performances of the different models on the five benchmark datasets, including Set5 [7], Set14 [60], BSD100 [40], Urban100 [22], and Manga109 [41]. We used PSNR (dB) and SSIM [44] scores on the Y channel of the YCbCr space as the metrics. LR images were acquired by the MATLAB bicubic kernel from corresponding HR images matching each SR task.

4.2. Comparisons of Super-Resolution Results

In Tab. 2, we compare NGswin with our other efficient SR models, including EDSR-baseline (CVRPRW17) [28], MemNet (ICCV17) [50], CARN (ECCV18) [2], IMDN (ACMMM19) [23], LatticeNet (ECCV20) [37], RFDN-L (ECCV20) [30], SRPN-Lite (ICLR22) [65], HNCT

2076
Table 4. Ablation study on the N-Gram context. The top and bottom tables are PSNR / SSIM of NGswin and HNCT, respectively.

| N-Gram | Direction | Type | Mult-Adds | #Params | Urban100 | Manga109 |
|--------|-----------|------|-----------|---------|----------|----------|
| w/o    | 2 tasks   |       |           |         |          |          |
| w/     | 2 tasks   |       |           |         |          |          |
| w/o    | 3 tasks   |       |           |         |          |          |
| w/     | 3 tasks   |       |           |         |          |          |
| (channel up) |       |       |           |         |          |          |
| (depth up)  |       |       |           |         |          |          |

Table 5. Ablation study on N-Gram interaction. “Direction”: how many directions the network see N-Gram neighbors from. “Type”: the method for N-Gram interaction. The bottom row is a default setting. PSNR / SSIM are evaluated on ×2 task with NGswin.

4.3. Ablation Studies

Tab. 4 demonstrates that the N-Gram context enhanced Swin Transformer-based SR models with a reasonable level of sacrificed efficiency. We denote HNCT [16] improved with the N-Gram context as HNCT-NG. The results of SwinIR-light [27] and a corresponding SwinIR-NG can be referred to in Tab. 3. Interestingly, the N-Gram increased SSIM in general. That is, our method tended to produce perceptually more similar images to the ground-truth. Moreover, our method was robust to Urban100 and Manga109 datasets, which are hard to recover with DIV2K training dataset [38]. Fig. 8 visualizes the examples. One can doubt the marginal gain of HNCT-NG. This was because HNCT contains 8 Swin Transformer layers (Swins), while NGswin had 20. The application of the N-Gram method on fewer Swins of HNCT-NG yielded less effective results. Afterwards, we increased the depth or channel of NGswin without N-Gram for more compelling comparisons regarding to the number of parameters. The variations were applied to ×4 task, which still fell behind our proposed NGswin de-
Figure 8. Visual comparisons of w/ vs. w/o N-Gram context for NGswin, SwinIR-light, and HNCT. At the top and bottom row, the 2nd to 4th columns show the models with and without N-Gram, respectively. More visual comparisons are in the supplementary Sec. D.

Table 6. Ablation study on extra stages and SCDP bottleneck.

(a) The specifications of models with different stages. dep.: # of NSTBs / res.: training input resolution. The total number of NSTBs is kept as 20.

| Stages | SCDP | Scale | Multi-Adds | #Params | Urban100 | Manga109 |
|--------|------|-------|------------|---------|---------|---------|
| extra  | w/o | × 2 | 87.39G | 99K | 32.28 / 0.9298 | 38.72 / 0.9777 |
| default| w/o | × 3 | 138.85G | 99K | 32.48 / 0.9321 | 38.92 / 0.9776 |
| extra  | w/ | | 140.41G | 99K | 32.53 / 0.9324 | 38.97 / 0.9777 |
| default| w/ | | 42.10K | 1.00K | 28.37 / 0.8562 | 33.97 / 0.9463 |
| extra  | w/o | × 4 | 65.85G | 1.00K | 28.47 / 0.8596 | 33.81 / 0.9464 |
| default| w/ | | 66.56G | 1.007K | 28.52 / 0.8603 | 33.89 / 0.9470 |
| extra  | w/o | × 3 | 36.06G | 1.013K | 26.38 / 0.7954 | 30.71 / 0.9121 |
| default| w/ | | 36.44G | 1.019K | 26.45 / 0.7963 | 30.80 / 0.9128 |

(b) Impacts of extra stages and SCDP bottleneck. PSNR / SSIM.

| Stages | SCDP | Scale | Multi-Adds | #Params | Urban100 | Manga109 |
|--------|------|-------|------------|---------|---------|---------|
| extra  | w/o | × 2 | 32.83G | 99K | 32.28 / 0.9298 | 38.72 / 0.9777 |
| default| w/o | × 3 | 28.17G | 99K | 32.48 / 0.9321 | 38.92 / 0.9776 |
| extra  | w/ | | 30.80G | 99K | 32.53 / 0.9324 | 38.97 / 0.9777 |
| default| w/ | | 42.10K | 1.00K | 28.37 / 0.8562 | 33.97 / 0.9463 |
| extra  | w/o | × 4 | 26.22G | 1.00K | 28.47 / 0.8596 | 33.81 / 0.9464 |
| default| w/ | | 27.25G | 1.013K | 26.38 / 0.7954 | 30.71 / 0.9121 |
| extra  | w/o | × 3 | 24.28G | 1.019K | 26.45 / 0.7963 | 30.80 / 0.9128 |
| default| w/ | | 24.66G | 1.025K | 26.52 / 0.7964 | 30.87 / 0.9130 |

5. Conclusion

This paper successfully introduced the N-Gram context from the text to the vision domain for the first time in history. Our N-Gram interaction by sliding-WSA made NGswin, SwinIR, and HNCT overcome the limitation of Swin Transformer, where the broad regions are ignored. SCDP bottleneck prevented the hierarchical encoder from dropping performance. The hierarchical encoder, small decoder, and uni-Gram embedding decreased the operations significantly. With the components above, NGswin showed competitive results compared with the previous leading SR methods. Moreover, SwinIR-NG established state-of-the-art results. For future works, we hope our N-Gram context can succeed on other low-level vision tasks, such as denoising and deblurring. In closing, if the N-Gram context is extended to the universal Transformer architectures, more developments for computer vision could be expected.

Acknowledgements. This paper was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant (No.2022-0-00956) and Korea Health Industry Development Institute (KHIDI) grant (No. H122C1983) funded by the Korea government (MSIT). Special thanks to Rescale Inc. for HPC cloud computing resources and Somang Kim for valuable comments.
ing swin transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 1833–1844, 2021. 1, 2, 4, 5, 6, 7

[28] 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. 2, 4, 6

[29] Zudi Lin, Prateek Garg, Atmadeep Banerjee, Salma Abdel Magid, Deqing Sun, Yulun Zhang, Luc Van Gool, Donglai Wei, and Hanspeter Pfister. Revisiting rcn: Improved training for image super-resolution. arXiv preprint arXiv:2201.11279, 2022. 6

[30] Jie Liu, Jie Tang, and Gangshan Wu. Residual feature distillation network for lightweight image super-resolution. In European Conference on Computer Vision, pages 41–55. Springer, 2020. 6

[31] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12009–12019, 2022. 2, 4

[32] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 10012–10022, 2021. 1, 2, 4

[33] Inigo Lopez-Gazpio, Montse Maritxalar, Mirella Lapata, and Eneko Agirre. Word n-gram attention models for sentence similarity and inference. Expert Systems with Applications, 132:1–11, 2019. 2, 3

[34] Ilya Loshchilov and Frank Hutter. Sgd: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983, 2016. 5

[35] Zhisheng Lu, Hong Liu, Juncheng Li, and Linlin Zhang. Efficient transformer for single image super-resolution. arXiv preprint arXiv:2108.11084, 2021. 1, 2, 4, 6, 7

[36] Xiaotong Luo, Yanyun Qu, Yuan Xie, Yulun Zhang, Cuihua Li, and Yun Fu. Lattice network for lightweight image restoration. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022. 7

[37] Xiaotong Luo, Yuan Xie, Yulun Zhang, Yanyun Qu, Cuihua Li, and Yun Fu. Latticenet: Towards lightweight image super-resolution with lattice block. In European Conference on Computer Vision, pages 272–289. Springer, 2020. 2, 6

[38] Salma Abdel Magid, Zudi Lin, Donglai Wei, Yulun Zhang, Jinjin Gu, and Hanspeter Pfister. Texture-based error analysis for image super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2118–2127, 2022. 7

[39] P Majumder, M Mitra, and BB Chaudhuri. N-gram: a language independent approach to ir and nlp. In International conference on universal knowledge and language, 2002. 2

[40] 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 Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001, volume 2, pages 416–423. IEEE, 2001. 6

[41] Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamashita, and Kiyoharu Aizawa. Sketch-based manga retrieval using manga109 dataset. Multimedia Tools and Applications, 76(20):21811–21838, 2017. 6

[42] Ben Niu, Weilei Wen, Wenqi Ren, Xiandge Zhang, Lianping Yang, Shuzhen Wang, Kaihao Zhang, Xiaochun Cao, and Haifeng Shen. Single image super-resolution via a holistic attention network. In European conference on computer vision, pages 191–207. Springer, 2020. 5

[43] Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. Unsupervised learning of sentence embeddings using compositional n-gram features. arXiv preprint arXiv:1703.02507, 2017. 2, 3

[44] Yatian Pang, Wenzhao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. arXiv preprint arXiv:2203.06604, 2022. 3, 4

[45] Russell Reed. Pruning algorithms—a survey. IEEE transactions on Neural Networks, 4(5):740–747, 1993. 2

[46] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention, pages 234–241. Springer, 2015. 2, 3, 4, 5

[47] Hans-Christian Ruiz-Euler, Unai Alegre-Ibarra, Bram van de Ven, Hajo Broersma, Peter A Bobbert, and Wilfred G van der Wiell. Dopant network processing units: towards efficient neural network emulators with high-capacity nanoelectronic nodes. Neuromorphic Computing and Engineering, 1(2):024002, 2021. 1

[48] Wenzhi Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1874–1883, 2016. 4, 5

[49] Yan Song, Tong Zhang, Yonggang Wang, and Kai-Fu Lee. Zen 2.0: Continue training and adaption for n-gram enhanced text encoders. arXiv preprint arXiv:2105.01279, 2021. 2

[50] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In Proceedings of the IEEE international conference on computer vision, pages 4539–4547, 2017. 6

[51] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang. Ntire 2017 challenge on single image super-resolution: Methods and results. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pages 114–125, 2017. 6

[52] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, Octo-
Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in neural information processing systems, 30, 2017.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing, 13(4):600–612, 2004.

Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 17683–17693, 2022.

Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal self-attention for local-global interactions in vision transformers. arXiv preprint arXiv:2107.00641, 2021.

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.

Qihang Yu, Yingda Xia, Yutong Bai, Yongyi Lu, Alan L Yuille, and Wei Shen. Glance-and-gaze vision transformer. Advances in Neural Information Processing Systems, 34:12992–13003, 2021.

Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In CVPR, 2022.

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.

Jiale Zhang, Yulun Zhang, Jinjin Gu, Yongbing Zhang, Linghe Kong, and Xin Yuan. Accurate image restoration with attention retractable transformer. arXiv preprint arXiv:2210.01427, 2022.

Qiming Zhang, Yufei Xu, Jing Zhang, and Dacheng Tao. Vsa: learning varied-size window attention in vision transformers. In Computer Vision--ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXV, pages 466–483. Springer, 2022.

Xindong Zhang, Hui Zeng, Shi Guo, and Lei Zhang. Efficient long-range attention network for image super-resolution. arXiv preprint arXiv:2203.06697, 2022.

Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In Proceedings of the European conference on computer vision (ECCV), pages 286–301, 2018.

Yulun Zhang, Huan Wang, Can Qin, and Yun Fu. Learning efficient image super-resolution networks via structure-regularized pruning. In International Conference on Learning Representations, 2021.