A Progressive Approach for Image Super-Resolution With Low-Resolution Increments

Anonymous Authors[1]

Abstract

A variety of Super-Resolution (SR) networks have recently been proposed, achieving increasingly better super-resolving performance. However, if the super-resolved image does not meet the requirements even with the best-performing SR module currently available, what can be done? We notice that, in the context of image transmission, if the super-resolved image fails to satisfy the receiver's requirements, it can be improved by retransmitting a larger low-resolution version from the sender. Unfortunately, this will result in redundant information transmission and bandwidth waste. In this paper, we propose a progressive approach for image SR that allows a sender to transmit low-resolution increments containing additional information to the receiver, which can be used to improve the super-resolved image on the basis of the previously received low-resolution version. To achieve this, we introduce a progressive SR framework, in which a convolutional structure is designed to equalize each down-sampled low-resolution pixel in its significance, enabling low-resolution information to be added when required. We further enhance our framework with a block-based strategy, allowing the information increment to focus only on a region of interest, thereby reducing transmission demand. Experiments and implementation validate the performance of the proposed progressive approach.

1. Introduction

Continuous improvement in camera resolutions has provided consumers with better visual experiences. However, even with high-performance compression algorithms, delivering a large number of large-sized high-resolution images across the Internet still places tremendous pressure on its core and access networks. Image SR addresses this issue (Liu et al., 2022; Qi et al., 2023) by allowing the sender to down-sample an image and transmit its low-resolution version, matching the targeted display device size. When necessary, the receiver can super-resolve it back to the original size using a specific SR method.

Various research efforts have aimed at improving the quality of super-resolved images. Early SR methods rely on multi-layer neural networks, surpassing traditional interpolation algorithms (Dong et al., 2014; Zhang et al., 2019). Recently, transformer and diffusion models are explored in image SR (Wang et al., 2023; Shang et al., 2024). With an end-to-end approach, image sampling can be included in the process of image SR. Joint SR models based on this are then investigated, leading to further improvements in image quality (Xu et al., 2023; Sun and Chen, 2024). If, however, the quality of the super-resolved image still does not meet user requirements, the sender may have to retransmit another low-resolution image of a larger size, containing more original information and more conducive to the super-resolving process, in the hope of leading to a higher quality super-resolved image. Unfortunately, this would result in duplicate data transmission, as the receiver had previously received a low-resolution version of the same image.

This paper explores the problem of image SR in the context of image transmission, aiming to allow the sender to incrementally transmit added down-sampled information to the receiver to further improve the super-resolved image on the basis of the previously received low-resolution version. However, leveraging increments to gradually improve the super-resolved image quality introduces a number of non-trivial challenges. First, when requested by the receiver for additional information, the sender must be able to continue sampling additional information, although a complete low-resolution version has previously been sampled. Second, existing SR methods at the receiver should be able to effectively leverage the increments to improve super-resolved images. Last but not the least, the receiver may only need enhancements for a certain region of the image, providing additional opportunities for super-resolving on-demand if an increment can focus only on the region of interest.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

We address these challenges by designing the progressive approach to image SR. We borrow the concept of classical convolution to generate down-sampled low-resolution information utilizing its randomized parameters. In this context, n channels of the convolution kernel correspond to n repeated sampling processes. As additional m low-resolution pixels are required, the convolution kernel is further expanded by m channels, providing the capability to add low-resolution information increments at arbitrary scales. The information increments, from multiple iterations of adding, are progressively integrated by our carefully designed processing unit to form a new low-resolution version to be super-resolved by an SR module. With block partition, the progressive integration can be done on a block-by-block basis, achieving on-demand super-resolving.

The contributions of this paper are summarized below.

- We design a convolution-based progressive SR framework. One convolution operation on an image is abstracted as a sampling of the image, outputting a low-resolution pixel. By increasing the number of channels of the convolution kernel, the amount of low-resolution pixel increases accordingly, enabling progressive information increment for super-resolving process.
- We investigate a block-based strategy, transforming the convolution operation from the entire image to image blocks, thereby making convolution-based down-sampling more feasible. Furthermore, this strategy enables further reduction of transmission costs as only local information details can be added per user interests.
- We evaluate the proposed progressive scheme with both extensive experiments and system implementation. Experimental results indicate that increments can be progressively integrated for image super-resolving and transmission cost is significantly reduced, while system implementation shows that our progressive approach is practical with fast-enough response time.

2. Related Work

Image SR has been introduced as a new approach to image transmission with reduced bandwidth requirements (Liu et al., 2022; Chen et al., 2024). SR-based image transmission involves the delivery of small-sized, low-resolution images to reduce the amount of transmitted data, where low-resolution images can be used directly and, if necessary, super-resolved up to the original size using certain SR methods (Yang et al., 2023; Yu et al., 2023; Zamfir et al., 2023).

Research on image SR has mainly focused on various SR network models, striving to enhance the quality of super-resolved images. The first network-based image SR method is presented in (Dong et al., 2014), and further improved in (Lai et al., 2017; Lim et al., 2017; Ahn et al., 2018) with the introduction of deeper residual neural networks. Generative adversarial networks are employed in (Zhang et al., 2019; Wang et al., 2021) to infer super-resolved images with visually realistic effect. Recently, transformer-based image SR methods are introduced by leveraging attention mechanisms (Wang et al., 2023; Chen et al., 2023; Yoo et al., 2023). Diffusion models have also been applied to image SR, leveraging diffusion to generate high-quality super-resolved images (Shang et al., 2024; Yue et al., 2024). Specifically for image transmissions, SR network can be jointly trained with a sampling network, and super-resolved images are further improved (Sun and Chen, 2020; Xu et al., 2023). Although advancements in SR networks can lead to higher-quality super-resolved images, their quality ultimately is constrained by the amount of information contained in the low-resolution images. In other words, under the same SR network, larger low-resolution images correspond to higher quality super-resolved images.

In classical Compressed Sensing (CS) theory (Donoho, 2006), an image can be compressed into measurements of any scale. From these measurements, an initial version of the same size as the original image is first reconstructed, and then its quality is improved through subsequent processing (Song et al., 2023; Zhang and Ghanem, 2018; Shen et al., 2024). In this paper, we will borrow the idea of CS reconstruction, first expanding the low-resolution image with increments, and then improving image quality with an SR module.

3. Methodology

When the quality of super-resolved images cannot meet user requirements, the sender can transmit *additional* low-resolution information increments that can be utilized by the SR module to enhance the super-resolved images. By doing so, the super-resolved images are gradually improved with each increment, while allowing the sender to terminate transmission at any point. This approach saves bandwidth for image transmissions while retaining full flexibility in image quality control. We term this scheme *progressive SR*.

3.1. Progressive SR Framework

The proposed progressive SR framework, illustrated in Fig. 1, comprises an initial and additional sampling units, the corresponding processing units, and a general SR module that can be any existing SR network.

Sampling Unit. The initial sampling unit $S\text{-}U_0$ and the additional sampling unit $S\text{-}U_i$ ($i \geq 1$) are executed at the sender. Let X denote the image to be transmitted, p_k be the

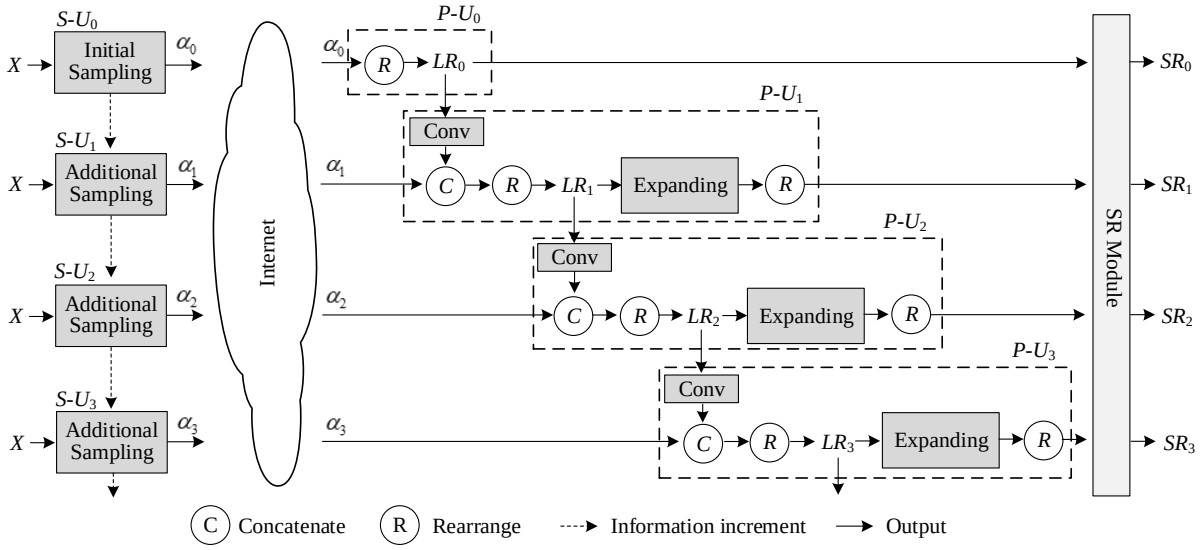


Figure 1. Progressive SR framework. α_0 and α_i represent the initial and i-th sampled low-resolution information, respectively. LR_0 is the visual version of α_0, SR_0 is the corresponding super-resolved image, and LR_i and SR_i denote the version after the i-th increment is concatenated.

k-th sampled low-resolution pixel, and ϕ be the sampling module. The sampling process is defined as

$$p_k = \mathcal{Q}(\phi(k) \otimes X), \tag{1}$$

where $\otimes$ represents the sampling operation, $\phi(k)$ denotes the k-th channel of ϕ, and $\mathcal{Q}(\cdot)$ is a quantization module that converts floating-point values into 8-bit unsigned int, since an image is commonly represented by using integer number ranging from 0 to 255. At the inception of the transmission of image X, the sender first transmits the image's initially-sampled information, i.e., α_0. Assuming that α_0 includes n_0 low-resolution pixels, α_0 can be formulated as

$$\alpha_0 = \{p_k\}_{k=1}^{n_0}. \tag{2}$$

If the sender receives a request for m_i additional pixels, unit $S\text{-}U_i$ is performed and m_i low-resolution pixels, denoted by increment α_i, are added. Mathematically, we have

$$\alpha_i = \{p_k\}_{k=n_{i-1}}^{n_i}, \tag{3}$$

where $n_i = n_{i-1} + m_i$. The process of sending bits in response to requests will be repeated until the user is satisfied.

Processing Unit. $P\text{-}U_0$ and $P\text{-}U_i$ ($i \geq 1$) represent the processing units that treat the down-sampled information of the original image transmitted from the corresponding sampling units, and the results will serve as the input of the subsequent SR module. With the Rearranging operation, α_0 is rearranged into its visual form, LR_0. If additional increment α_1 is available, through the Conv module $conv(\cdot)$, the feature of LR_0 is concatenated with the newly available increment α_1, and the result is rearranged into LR_1. This process of concatenating and rearranging can be executed repeatedly until the user demand is met. Presented mathematically,

$$LR_0 = \mathcal{R}(\alpha_0), \tag{4}$$

$$LR_i = \mathcal{R}(\mathcal{C}(\alpha_i, conv(LR_{i-1}))), \tag{5}$$

where $\mathcal{C}(\cdot,\cdot)$ and $\mathcal{R}(\cdot)$ denote the operations of Concatenating and Rearranging, respectively. The Expanding module is responsible for expanding LR_i to a suitable size to match the subsequent SR module, since most SR methods remain integral-factor super-resolving.

SR Module. The SR module can be a network structure of any existing SR method.

Units $S\text{-}U_i$, $P\text{-}U_i$, and SR module form the proposed progressive SR framework. Let LR and HR represent the ground truth of low-resolution and original high-resolution image, respectively. The loss function, $\mathcal{L}$, is defined as

$$\mathcal{L} = \lambda_1 \cdot mse(LR_i, LR) + \lambda_2 \cdot l_1(SR_i, HR), \tag{6}$$

where mse and l_1 denote mean squared error and standard l_1 norm loss, respectively, λ_1 and λ_2 are trade-off hyper-parameters that balance these two loss terms. Note that more sophisticated feature-level perception loss can be employed to improve performance. The proposed framework is trained utilizing an end-to-end strategy. Using Eq. (6) with $i = 0$, units $S\text{-}U_0$, $P\text{-}U_0$, and SR module are first jointly trained. Subsequently, $S\text{-}U_0$ is fixed, and $S\text{-}U_1$, $P\text{-}U_1$, and SR are further jointly trained using Eq. (6) with $i = 1$. This process is repeated until all $S\text{-}U_i$ and $P\text{-}U_i$ are trained.

3.2. Block-Based Design

This subsection designs the sampling module ϕ by exploiting block partition strategy. The Expanding module, Quantization module $\mathcal{Q}(\cdot)$, Conv module, and operations of Concatenating and Rearranging are also introduced.

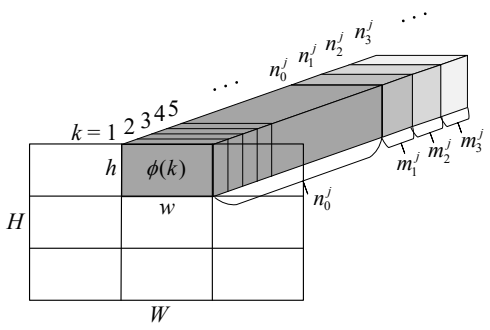


Figure 2. Block-based design of progressive sampling module.

Fig. 2 illustrates the block-based progressive sampling module, which is based on the classical convolution kernel structure, where multiple channels are borrowed to achieve progressivity. Intuitively, to obtain a complete sampling of the entire image, the convolution kernel must be designed to be of size $H \times W$ that matches the height and width of the image X. In this way, one convolution operation, $\phi(k) \otimes X$, is comparable with one sampling procedure of image X. Here, $\phi(k)$ denotes the k-th channel of the convolution kernel ϕ and $\otimes$ represents the convolution operation. However if X is a high-resolution image, employing a convolution structure to match its size can result in excessively large convolution kernels. Given this observation, we present a block-based alternative. The size of the convolution kernel on the k-th channel, $\phi(k)$, is designed to be of size $h \times w$, size of one block of image X. n_0^j and m_i^j denote the amount of initial and the additional i-th sampled increment for the j-th block, respectively, and $n_i^j = n_{i-1}^j + m_i^j$. Suppose that image X is partitioned into J blocks, where $J = rs$, $r = \frac{H}{h}$, $s = \frac{W}{w}$, and B_j denotes the j-th block. One sampling operation outputs a low-resolution pixel, p_k^j, satisfying $p_k^j = \mathcal{Q}(\phi(k) \otimes B_j)$. Then, $\alpha_0 = \{\{p_k^j\}_{k=1}^{n_0^j}\}_{j=1}^{J}$, while α_i can be represented as

$$\alpha_i = \{\{p_k^j\}_{k=n_{i-1}^j}^{n_i^j}\}_{j=1}^{J}. \tag{7}$$

Block-based design avoids overwhelming convolution operations, making the sampling process practical. Additionally, users are often interested only in specific regions of an image, and block-based design allows for super-resolving on demand.

The Expanding module is responsible for expanding low-resolution LR_i ($i \geq 1$) to the smallest integral multiple of initial LR_0 to match the SR module. This is because most existing SR methods are based on an integer super-resolving factor, and we set the expanding factor to the smallest integer multiple of LR_0 with the intention of reducing the computational cost of the SR module. The Expanding module is designed as the inverse process of the Sampling module, achieved by using a block-based convolution structure as well. Suppose that one block of image X is of size $h \times w$, and Φ^i denotes the block-based convolution kernel in unit $P\text{-}U_i$. Then, Φ^i is of size n_i and its channel number is c, where $c = \frac{hw}{l^2}$ and l is the super-resolving factor. The output of the Expanding module for j-th low-resolution block of LR_i is $\{P_k^j\}_{k=1}^{c}$, where $P_k^j = \Phi^i(k) \otimes b_j$, b_j represents j-th block of LR_i. These c pixels are rearranged to a size of $(\frac{h}{l}) \times (\frac{w}{l})$ which is used as input of the SR module with a super-resolving factor of l.

Quantization module $\mathcal{Q}(\cdot)$ employs the soft round strategy (Nakanishi et al., 2019) to derive the gradient during the back-propagation process of the training phase, since a simple rounding operation that rounds the sampled result to its nearest integer number is not differentiable. The Conv module consists of a general convolutional layer, $conv(\cdot)$, which is used to match visual LR_i to down-sampled randomized coefficient α_{i+1}. Concatenating operation $\mathcal{C}(\cdot,\cdot)$ adds additional increments to the previous sampled information, while the Rearranging operation $\mathcal{R}(\cdot)$ arranges the concatenated information to an image format. Note that in unit $P\text{-}U_i$, the second Rearranging operation further arranges the individual image blocks into a complete image before entering them into the SR module. In this way, blocking artifacts can be ameliorated.

3.3. Progressive Transmission Protocol

This section exemplifies a simple image transmission protocol to implement the proposed progressive SR framework, as described in Algorithm 1. The sender first delivers a low-resolution version of the image, α_0, to the receiver, which is then super-resolved. Based on a stopping crite-

Algorithm 1 Transmission Protocol

Input: SR_0
Output: *stop*: true/false

$i \leftarrow 0$, *stop* $\leftarrow$ false;
while !*stop* **do**
  if SR_i satisfies the requirements **then**
    stop $\leftarrow$ true;
  else
    $i \leftarrow i + 1$;
    $LR_i \leftarrow \mathcal{R}(\mathcal{C}(\alpha_i, conv(LR_{i-1})))$;
  end if
end while
return *stop*;

rion, the receiver evaluates whether the resolved image, SR_0, meets the requirements. If not, it then requests the sender to transmit an additional low-resolution increment, α_1, and evaluates SR_1 again. This process repeats until the receiver's requirements are met.

Note that we are only presenting the basic concept of the protocol, and various implementation details are omitted. For example, the receiver can assess whether SR_i satisfies the requirements based on its visual effect or by using a non-reference metric such as PI (Blau et al., 2018).

4. Experimental Result

This section conducts extensive experiments to evaluate the proposed progressive SR approach. CNN-based SR networks EDSR (Lim et al., 2017) and CARN (Ahn et al., 2018), and Transformer-based SR networks Omni (Wang et al., 2023) are adopted as SR modules in our framework. It is important to note that any other SR networks can also be used in a similar manner. For initial sampling transmission, the super-resolving factor is set to $\times 4$, while for additional incremental sampling transmission, the expanding factor and the super-resolving factor are set to $\times 2$.

4.1. Experimental Setup

Datasets and Metrics. The popular DIV2K and Flickr2K are used as training datasets. To accelerate image loading during training, each training image is evenly divided into four parts, with each part becoming a separate, relatively smaller training image. For testing purpose, we use five standard benchmark datasets, including Set5, Set14, BSDS100, Urban100, and the validation set of DIV2K. PSNR and SSIM are adopted to evaluate the quality of the super-resolved image on the Y channel of the transformed YCbCr space.

Training Details. Training images are augmented with random flips and 90-degree rotations. The ground truth of $\times 4$ low-resolution version is generated by bicubic interpolation algorithm from the corresponding original image. AdamW optimizer (Loshchilov and Hutter, 2019) is employed to train the proposed framework with a batch size of 16 for 300K iterations. The initial learning rate is set to 0.0005 and halved for every 50K iterations. In each training batch, the original high-resolution images are randomly cropped into patches of size 64×64 to be used as input. All experiments are conducted by employing PyTorch 2.2.0 on a personal computer configured with one GPU NVIDIA GeForce RTX 4090, one CPU Intel Core i7-14700KF with frequency 3.40 GHz, and 64 GB of memory.

Block Size. The sampling modules employ block-based multi-channel convolution structure. Block size is set to 32×32 in our experiments, except in Tab. 2, where various block sizes are compared.

4.2. Transmission Cost

The proportion of down-sampled low-resolution information compared to the original image is used to estimate the transmission cost. For a fair comparison, both the SR-based transmission and the SR module in our progressive approach use the same SR network. Here we only show the results of Omni due to space constraints. Note that EDSR and CARN have similar results.

Fig.3 illustrates local information increment for the proposed progressive approach, using the 0817th image from the DIV2K validation set as an example. The sizes of the regions of interest are set to 9, 25, and 64 blocks. It can be seen that, employing block-based local information increments, our approach transmits significantly less data. This is because, when transmitting additional information, the sender sends only the information of the corresponding region of interest rather than the entire image.

Fig. 4 evaluates our progressive approach on five benchmark datasets. For each dataset, the first point on the left indicates the transmission cost of the initial low-resolution version, α_0. The two methods coincide as expected. It can be seen that, with the same quality of the super-resolved image, our approach incurs obviously less transmission cost. Furthermore, as the super-resolved images are enhanced, our method saves more on transmission cost. Here, we use SSIM to represent image quality. PSNR yields similar results.

4.3. Quality of Super-resolved Image

Tab. 1 shows that as down-sampled low-resolution increments are added, the average values of PSNR and SSIM of the five datasets gradually increase. This indicates that, whether EDSR, CARN, or Omni is used for the SR module, our proposed progressive SR approach consistently works as expected. In the experiments, for each 8×8 low-resolution

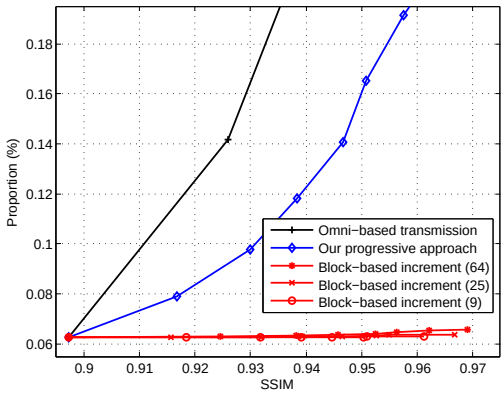
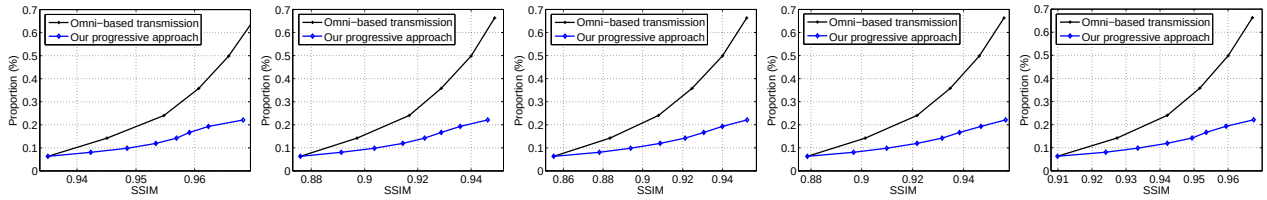


Figure 3. Significantly less transmission cost by employing block-based local information increment.



(a) Set5 (b) Set14 (c) BSDS100 (d) Urban100 (e) DIV2K

Figure 4. Less transmission cost for our progressive approach in terms of percentage of low-resolution information to original image.

image block, the number of pixels included in increments α_1-α_7 is set to 17, 19, 21, 23, 25, 27, and 29, respectively.

Tab. 2 depicts the impact of various block sizes on the quality of the super-resolved image. Here, the scenarios of initial sampling α_0, increments α_2, and α_4 are taken as examples. It can be observed that, with the same increments, the values of PSNR and SSIM obtained from the three block sizes are similar. That is, the impact of sizes of different convolution blocks on image quality is not significant.

In Fig. 5, using the 0817th low-resolution image from the DIV2K as an example, we demonstrate the visual effects of the proposed progressive approach by gradually adding information to the super-resolved images. As the low-resolution information increments are added from α_1 to α_7, the visual effect gradually becomes clearer. This information adding process allows for termination whenever satisfied, thereby saving transmission cost.

Fig. 6 illustrates how information is added by block. Fig 6(a) is the 0831st low-resolution image from DIV2k, where the user is interested in the content within the right red box, needing to recognize the numbers, while not interested in the content within the left green box. With block-based increment, we can add information only for pixels in the red box. From the super-resolved image shown in Fig. 6(b), block-based increment achieves the expected result, with the numbers '67' in the red box being clearly displayed. Note that almost no blocking artifacts can be observed in the super-resolved image.

4.4. Ablation Study

Tab. 3 investigates the impact with and without the Conv layer in the proposed framework using the DIV2K validation set and the Omni module. It can be observed that the Conv layer offers a slight improvement in terms of PSNR and SSIM. The reason might be that, by using the Conv layer, the concatenating operation is performed at the feature level rather than at the pixel level, thus allowing for better learning of information beneficial for image super-resolving.

Fig. 7 explores the impact of two training strategies, where one is to train a unified SR module for the entire framework, and the other is to train a specific SR model separately each time when information is added. It can be seen that unified SR module has almost the same PSNR values as separately-trained ones. This shows that the unified training strategy can be adopted without the need to train a separate SR module for each information increment.

5. Implementation

We implement progressive SR in image transmission scenario on Internet. Units S-U_0 and S-U_i ($1 \leq i \leq 7$) are deployed at a server that acts as a sender, while the receiver runs on a laptop for image super-resolving by running P-U_0, P-U_i, and SR modules EDSR, CARN, and Omni.

Sender Side. The sender is carried out from a traditional CPU server platform configured with two CPUs @2.1G and 16GB of memory. It is developed with the Flask Web application framework and executes Python codes to respond to HTTP requests from the receiver.

Receiver Side. The receiver runs on a laptop configured with one NVIDIA's Geforce RTX 4080 laptop GPU, one CPU @2.2G, and 32GB of memory. Through a user interface, the user first selects the image that needs to be downloaded. The receiver then sends an HTTP request to the server for an initial low-resolution version of the image. Next, the receiver uses processing unit and SR module to generate the super-resolved image that is displayed in the user interface. At this point, according to the transmission protocol, if the user continues to request additional increments, the previous process is repeated.

Tab. 4 indicates that the response time remains very fast, at the level of 0.1 seconds. Here, the response time is for transmitting a high-resolution 2K image with local information increment shown in Fig. 6 of Sec. 4.3. The sender's response time refers to the duration from receiving a request to transmitting down-sampled information. The receiver's time refers to that taken from receiving the down-sampled information to when the super-resolved image is generated. The data represent the average of 10 repeated transmissions, since the response time can be affected by network condition. Note that the response time for the initial request

SR Module	Set5	Set14	BSDS100	Urban100	DIV2K
EDSR (α_0)	34.83 / 0.9385	30.87 / 0.8855	30.46 / 0.8679	29.91 / 0.9025	33.68 / 0.9215
Increment α_1	35.29 / 0.9428	31.52 / 0.8935	30.91 / 0.8788	30.33 / 0.9090	34.18 / 0.9284
Increment α_2	35.77 / 0.9469	32.06 / 0.9027	31.41 / 0.8911	30.87 / 0.9172	34.72 / 0.9354
Increment α_3	36.33 / 0.9515	32.60 / 0.9116	31.95 / 0.9028	31.52 / 0.9265	35.31 / 0.9423
Increment α_4	36.78 / 0.9546	33.09 / 0.9189	32.51 / 0.9139	32.08 / 0.9336	35.87 / 0.9483
Increment α_5	37.22 / 0.9577	33.60 / 0.9257	33.02 / 0.9227	32.67 / 0.9404	36.39 / 0.9532
Increment α_6	37.69 / 0.9607	34.19 / 0.9326	33.63 / 0.9323	33.32 / 0.9473	37.02 / 0.9585
Increment α_7	38.26 / 0.9638	34.79 / 0.9386	34.24 / 0.9406	33.94 / 0.9529	37.64 / 0.9631
CARN (α_0)	34.36 / 0.9353	30.36 / 0.8778	30.17 / 0.8617	28.93 / 0.8861	33.22 / 0.9158
Increment α_1	34.95 / 0.9404	31.11 / 0.8879	30.67 / 0.8742	29.61 / 0.8980	33.81 / 0.9244
Increment α_2	35.49 / 0.9452	31.62 / 0.8974	31.16 / 0.8866	30.10 / 0.9070	34.35 / 0.9318
Increment α_3	36.05 / 0.9497	32.15 / 0.9068	31.68 / 0.8989	30.79 / 0.9181	34.94 / 0.9392
Increment α_4	36.53 / 0.9534	32.70 / 0.9154	32.25 / 0.9105	31.52 / 0.9280	35.54 / 0.9458
Increment α_5	36.87 / 0.9560	33.11 / 0.9213	32.69 / 0.9189	32.05 / 0.9350	36.00 / 0.9506
Increment α_6	37.35 / 0.9590	33.67 / 0.9280	33.32 / 0.9291	32.65 / 0.9416	36.62 / 0.9561
Increment α_7	37.81 / 0.9619	34.20 / 0.9339	33.88 / 0.9374	33.07 / 0.9467	37.16 / 0.9606
Omni (α_0)	34.47 / 0.9350	30.73 / 0.8760	30.06 / 0.8552	28.84 / 0.8786	32.97 / 0.9099
Increment α_1	34.89 / 0.9423	31.27 / 0.8913	30.68 / 0.8782	29.41 / 0.8968	33.54 / 0.9240
Increment α_2	35.75 / 0.9485	32.01 / 0.9037	31.39 / 0.8939	30.25 / 0.9098	34.33 / 0.9334
Increment α_3	36.32 / 0.9534	32.62 / 0.9143	32.06 / 0.9086	30.98 / 0.9217	35.04 / 0.9421
Increment α_4	36.79 / 0.9569	33.15 / 0.9225	32.73 / 0.9212	31.69 / 0.9317	35.70 / 0.9493
Increment α_5	37.09 / 0.9591	33.57 / 0.9288	33.30 / 0.9306	32.19 / 0.9385	36.12 / 0.9535
Increment α_6	37.50 / 0.9624	34.13 / 0.9358	33.94 / 0.9401	32.89 / 0.9470	36.75 / 0.9593
Increment α_7	38.82 / 0.9683	35.38 / 0.9461	35.16 / 0.9524	34.18 / 0.9566	38.11 / 0.9674

Table 1. Gradual PSNR/SSIM improvement using low-resolution information increments α_i, $1 \leq i \leq 7$.

Block	Set5	Set14	BSDS100	Urban100	DIV2K
32×32 (α_0)	34.47 / 0.9350	30.73 / 0.8760	30.06 / 0.8552	28.84 / 0.8786	32.97 / 0.9099
16×16	34.75 / 0.9399	30.94 / 0.8834	30.16 / 0.8602	29.21 / 0.8892	33.06 / 0.9134
8×8	34.58 / 0.9396	30.93 / 0.8830	30.06 / 0.8586	29.56 / 0.8950	33.07 / 0.9131
32×32 (α_2)	35.75 / 0.9485	32.01 / 0.9037	31.39 / 0.8939	30.25 / 0.9098	34.33 / 0.9334
16×16	35.74 / 0.9502	32.10 / 0.9080	31.35 / 0.8935	30.34 / 0.9105	33.80 / 0.9240
32×32 (α_4)	36.79 / 0.9569	33.15 / 0.9225	32.73 / 0.9212	31.69 / 0.9317	35.70 / 0.9493
8×8	36.24 / 0.9526	33.06 / 0.9112	31.77 / 0.9003	31.29 / 0.9199	34.02 / 0.9172

Table 2. Comparison of PSNR/SSIM with various block sizes, which indicates the size of different convolution blocks does not have a significant impact on image quality. Here, Omni is used for the SR module.

Model	PSNR		SSIM	
	w/o conv	w/ conv	w/o conv	w/ conv
Omni (α_0)	32.97	32.97	0.9099	0.9099
Increment α_1	33.33	33.54	0.9180	0.9240
Increment α_2	34.15	34.33	0.9286	0.9334
Increment α_3	34.89	35.04	0.9385	0.9421
Increment α_4	35.34	35.70	0.9443	0.9493
Increment α_5	36.01	36.12	0.9521	0.9535
Increment α_6	36.50	36.75	0.9575	0.9593
Increment α_7	36.88	38.11	0.9614	0.9674

Table 3. Ablation study with and without Conv layer.

	Sender	Receiver		
		EDSR	CARN	Omni
Initial request	0.48	1.20	0.37	0.56
1^{st} Increment	0.36	0.53	0.07	0.12
2^{nd} Increment	0.37	0.47	0.06	0.11
3^{rd} Increment	0.37	0.46	0.06	0.11
4^{th} Increment	0.38	0.45	0.07	0.12
5^{th} Increment	0.38	0.48	0.07	0.13
6^{th} Increment	0.38	0.47	0.06	0.15
7^{th} Increment	0.38	0.46	0.07	0.12

Table 4. Response time of the sender and receiver (second).

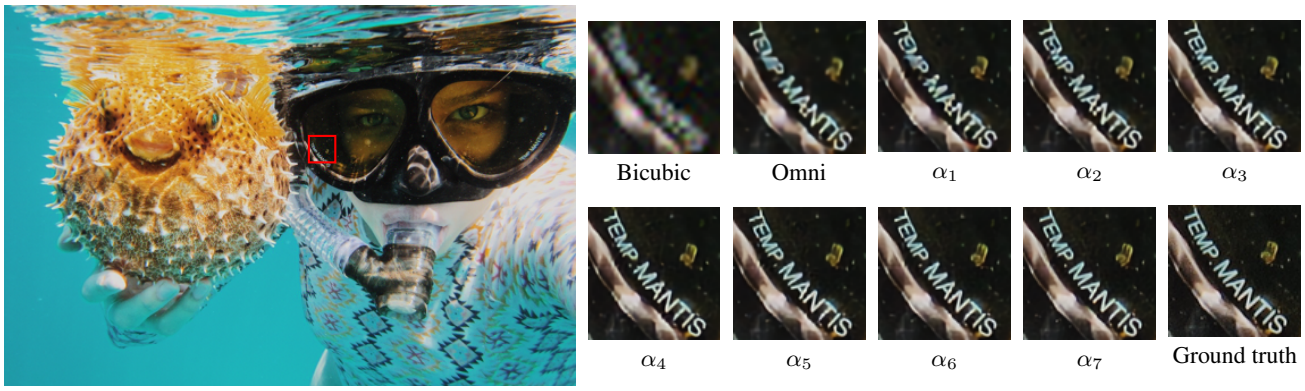


Figure 5. Visual effects that are gradually improved along with continual low-resolution information increments.

(a) Low-resolution image (b) Super-resolved image

Figure 6. Visual results by only adding local low-resolution information inside the red box. For the super-resolved image, the region inside the red box on the right is significantly clearer than the one inside the green box on the left.

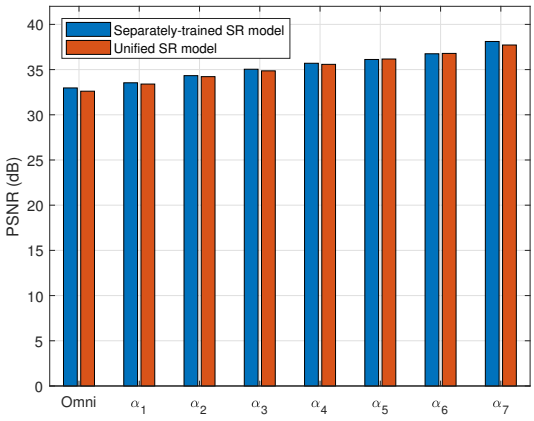


Figure 7. Ablation study with one unified and several separately-trained SR modules on the validation set of DIV2K.

is needed for all SR-based schemes, while the time for incremental requests is the time related to our progressive transmission. Also note that 7 additional increment requests are not required, and transmission can be halted at any time once the super-resolved image meets the requirements.

6. Conclusion

In this paper, we propose a progressive approach for image SR, allowing the sender to send low-resolution information increments in order to satisfy additional requirements from the receiver. The low-resolution increments can be gradually sent by the sender and continually utilized by the SR module at the receiver to progressively enhance the super-resolved images, avoiding the need to retransmit another low-resolution version of the image redundant with the initial one. Experimental results validate the progressivity of the proposed approach, and the system implementation indicates that our approach is practically feasible.

References

Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *Proceedings of the European conference on computer vision (ECCV)*, pages 252–268, 2018.

Yochai Blau, Roey Mechrez, Radu Timofte, Tomer Michaeli, and Lihi Zelnik-Manor. The 2018 pirm challenge on perceptual image super-resolution. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 1–22, 2018.

Xiangyu Chen, Xintao Wang, Jiantao Zhou, Yu Qiao, and Chao Dong. Activating more pixels in image super-resolution transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22367–22377, 2023.

Zheng Chen, Zongwei Wu, Eduard Zamfir, Kai Zhang, Yulun Zhang, Radu Timofte, Xiaokang Yang, Hongyuan Yu, Cheng Wan, Yuxin Hong, et al. Ntire 2024 challenge on image super-resolution (x4): Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6108–6132, 2024.

Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 184–199. Springer, 2014.

David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 624–632, 2017.

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 (CVPRW)*, pages 136–144, 2017.

Anran Liu, Yihao Liu, Jinjin Gu, Yu Qiao, and Chao Dong. Blind image super-resolution: A survey and beyond. *IEEE transactions on pattern analysis and machine intelligence*, 45(5): 5461–5480, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations*, 2019.

Ken M Nakanishi, Shin-ichi Maeda, Takeru Miyato, and Daisuke Okanohara. Neural multi-scale image compression. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part VI 14*, pages 718–732. Springer, 2019.

Chenyang Qi, Xin Yang, Ka Leong Cheng, Ying-Cong Chen, and Qifeng Chen. Real-time 6k image rescaling with rate-distortion optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14092–14101, 2023.

Shuyao Shang, Zhengyang Shan, Guangxing Liu, LunQian Wang, XingHua Wang, Zekai Zhang, and Jinglin Zhang. Resdiff: Combining cnn and diffusion model for image super-resolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8975–8983, 2024.

Minghe Shen, Hongping Gan, Chunyan Ma, Chao Ning, Hongqi Li, and Feng Liu. Mtc-csnet: Marrying transformer and convolution for image compressed sensing. *IEEE Transactions on Cybernetics*, 2024.

Jiechong Song, Chong Mou, Shiqi Wang, Siwei Ma, and Jian Zhang. Optimization-inspired cross-attention transformer for compressive sensing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6174–6184, 2023.

Wanjie Sun and Zhenzhong Chen. Learned image downscaling for upscaling using content adaptive resampler. *IEEE Transactions on Image Processing*, 29:4027–4040, 2020.

Wanjie Sun and Zhenzhong Chen. Learning many-to-many mapping for unpaired real-world image super-resolution and downscaling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Hang Wang, Xuanhong Chen, Bingbing Ni, Yutian Liu, and Jinfan Liu. Omni aggregation networks for lightweight image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22378–22387, 2023.

Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-ESRGAN: Training real-world blind super-resolution with pure synthetic data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1905–1914, 2021.

Bingna Xu, Yong Guo, Luoqian Jiang, Mianjie Yu, and Jian Chen. Downscaled representation matters: Improving image rescaling with collaborative downscaled images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12237–12247, 2023.

Jinhai Yang, Mengxi Guo, Shijie Zhao, Junlin Li, and Li Zhang. Self-asymmetric invertible network for compression-aware image rescaling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3155–3163, 2023.

Jinsu Yoo, Taehoon Kim, Sihaeng Lee, Seung Hwan Kim, Honglak Lee, and Tae Hyun Kim. Enriched cnn-transformer feature aggregation networks for super-resolution. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4956–4965, 2023.

Qian Yu, Qing Li, Rui He, Gareth Tyson, Wanxin Shi, Jianhui Lv, Zhenhui Yuan, Peng Zhang, Yulong Lan, and Zhicheng Li. Bisr: Bidirectionally optimized super-resolution for mobile video streaming. In *Proceedings of the ACM Web Conference 2023*, pages 3121–3131, 2023.

Zongsheng Yue, Jianyi Wang, and Chen Change Loy. Resshift: Efficient diffusion model for image super-resolution by residual shifting. *Advances in Neural Information Processing Systems*, 36, 2024.

Eduard Zamfir, Marcos V Conde, and Radu Timofte. Towards real-time 4k image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1522–1532, 2023.

Jian Zhang and Bernard Ghanem. Ista-net: Interpretable optimization-inspired deep network for image compressive sensing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1828–1837, 2018.

Wenlong Zhang, Yihao Liu, Chao Dong, and Yu Qiao. RankSRGAN: Generative adversarial networks with ranker for image super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV*, pages 3096–3105, 2019.