

# Learned Image Downscaling for Upscaling Using Content Adaptive Resampler

Wanjie Sun<sup>ID</sup> and Zhenzhong Chen<sup>ID</sup>, *Senior Member, IEEE*

**Abstract**—Deep convolutional neural network based image super-resolution (SR) models have shown superior performance in recovering the underlying high resolution (HR) images from low resolution (LR) images obtained from the predefined downscaling methods. In this paper, we propose a learned image downscaling method based on content adaptive resampler (CAR) with consideration on the upscaling process. The proposed resampler network generates content adaptive image resampling kernels that are applied to the original HR input to generate pixels on the downsampled image. Moreover, a differentiable upscaling (SR) module is employed to upscale the LR result into its underlying HR counterpart. By back-propagating the reconstruction error down to the original HR input across the entire framework to adjust model parameters, the proposed framework achieves a new state-of-the-art SR performance through upscaling guided image resamplers which adaptively preserve detailed information that is essential to the upscaling. Experimental results indicate that the quality of the generated LR image is comparable to that of the traditional interpolation based method and the significant SR performance gain is achieved by deep SR models trained jointly with the CAR model. The code is publicly available on: <https://github.com/sunwj/CAR>.

**Index Terms**—Image resizing, downscaling, super-resolution, content adaptive, non-uniform resampling.

## I. INTRODUCTION

AS THE smartphone cameras are starting to rival or beat DSLR cameras, a large number of ultra high resolution images are produced everyday. However, it is always reduced from its original resolution to smaller sizes that are fit to the screen of different mobile devices and web applications. Thus, it is desirable to develop an efficient image downscaling and upscaling method to make such application more practical and resources saving by only generating, storing and transmitting a single downsampled version for preview and upscaling it to high resolution when details are going to be viewed. Besides, the pre-downscaling and post-upscaling operation also helps to save storage and bandwidth for image or video compression and communication [1]–[4].

Image downscaling is one of the most common image processing operations, aiming to reduce the resolution of the

Manuscript received October 23, 2019; revised December 23, 2019; accepted January 20, 2020. Date of current version February 6, 2020. This work was supported in part by the National Natural Science Foundation of China under Contract 61771348. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Xin Li. (*Corresponding author: Zhenzhong Chen*)

The authors are with the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China (e-mail: zzchen@ieee.org).

This article has supplementary downloadable material available at <http://ieeexplore.ieee.org>, provided by the author.

Digital Object Identifier 10.1109/TIP.2020.2970248

high-resolution (HR) image while keeping its visual appearance. According to the Nyquist-Shannon sampling theorem [5], it is inevitable that high-frequency content will get lost during the sample-rate conversion. Contrary to the image downscaling task is the image upscaling, also known as resolution enhancement or super-resolution (SR), trying to recover the underlying HR image of the LR input. Image SR is essentially an ill-posed problem because an undersampled image could refer to numerous HR images. The quality of the SR result is very limited due to the ill-posed nature of the problem and the lost frequency components cannot be well-recovered [6], [7]. Previous work regards image downscaling and SR as independent tasks. Image downscaling techniques pay much attention to enhance details, such as edges, which helps to improve human visual perception [3]. On the other hand, recent state-of-the-art deep SR models have witnessed their capability to restore HR image from the LR version downsampled using traditional filtering-decimation based methods with great performance gain [8], [9]. However, the predetermined downscaling operations may be sub-optimal to the SR task and state-of-the-art deep SR models still cannot well recover fine details from distorted textures caused by the fixed downscaling operations.

In this paper, we proposed a learned content adaptive image downscaling model in which an SR model tries to best recover HR images while adaptively adjusting the downscaling model to produce LR images with potentially detailed information that are key to the optimal SR performance. The downscaling model is trained without any LR image supervision. To make sure that the LR image produced by our downscaling model is a valid image, we propose to employ the resampling method where content adaptive non-uniform resampling kernels predicted by a convolutional neural network (CNN) are applied to the original HR image to generate pixels of the LR output. Quantitative and qualitative experimental results illustrate that LR images produced by the proposed model can maintain comparable visual quality as the widely used bicubic interpolation based image downscaling method while advanced SR image quality is obtained using state-of-the-art deep SR models trained with LR images generated from the proposed CAR model.

Our contributions are concluded as follows:

- A learned image downscaling model is proposed which is trained under the guidance of the SR model. The proposed image downscaling model produces images that can be well super-resolved while comparable visual quality can

be maintained. Experimental results indicate a new state-of-the-art SR performance with the proposed end-to-end image downscaling and upscaling framework.

- The resampling method is employed to downscale image by applying content adaptive non-uniform resampling kernels on the original HR input, which can effectively maintain the structure of the HR input in an unsupervised manner. Because directly predicting the LR image by combining low and high-level abstract image features can not guarantee that the generated result is a genuine image without any LR image supervision.
- The learned content adaptive non-uniform resampling kernels perform non-uniform sampling and also make the size of resampling kernels to be more effective. The generated kernels produced by the proposed CAR model are composed of weights and sampling position offsets in both horizontal and vertical directions, making the learned resampling kernels adaptively change their weights and shape according to its corresponding resampling contents.

The rest of the paper is organized as follows. Section II reviews task independent and task driven image downscaling algorithms. Section III introduces the proposed SR guided content adaptive image downscaling framework, and the computing process of each component in the framework is explained. Section IV evaluates and analyzes experimental results for the SR images and downsampled images quantitatively and qualitatively. Finally, Section V summarizes our work.

## II. RELATED WORK

This section presents a review of image downscaling techniques aiming to maintain the visual quality of the LR image. Image downscaling algorithms can be categorized into two groups as follows.

### A. Task Independent Image Downscaling

Earlier work of image downscaling primarily tends to prevent aliasing [5] which arises during sampling rate reduction. Those methods are based on linear filters [10], where the HR image is firstly convolved with a low-pass kernel to push frequency components of the image below the Nyquist frequency [11], then being sub-sampled into target size. Many frequency-based filters are developed, *e.g.*, the box, bilinear and bicubic filter [12]. However, the downsampled images tend to be blurred because the high-frequency details are suppressed. Filters that are designed to model the ideal *sinc* filter, *e.g.*, the Lanczos filter [13], tend to produce ringing artifacts near strong image edges. All of these filters are predetermined with some of them having tuning parameters. The same filter is applied globally to the input HR image, ignoring the characteristics of image content with varying details.

Recently, many researchers begin to focus on the aspects of detail preserving and human perception when developing image downscaling algorithm. Kopf *et al.* firstly proposed a novel content adaptive image downscaling method based on a joint bilateral filter [14]. The key idea is to optimize the shape and locations of the downsampling kernels to better

align with local image features by considering both spatial and color variances of the local region. Öztireli and Gross [15] proposed a method to downscale HR images without filtering. They consider image downscaling as an optimization problem and use the structural similarity index (SSIM) [16] as objective to directly optimize the downsampled image against its original image. This approach helps to capture most of the perceptually important details. Weber *et al.* [17] proposed an image downscaling algorithm aiming to preserve small details of the input image, which are often crucial for a faithful visual impression. The intuition is that small details transport more information than bigger areas with similar colors. To that end, an inverse bilateral filter is used to emphasize differences rather than punishing them. Gastal and Oliveira [18] introduced the spectral remapping algorithm to control aliasing during image downscaling. Instead of discarding high-frequency information, it remaps such information into the representable range of the downsampled spectrum. Recently, Liu *et al.* [19] proposed a L0-regularized optimization framework for image downscaling, which is composed of a gradient-ratio prior and reconstruction prior. The downscaling problem is solved by iteratively optimize the two priors in an alternative way.

### B. Task Specific Image Downscaling

Most image downscaling algorithms only care about the visual quality of the downsampled image, so that the downsampled image may not be optimal to other computer vision tasks. To tackle this problem, task guided image downscaling has emerged. Zhang *et al.* [3] took the quality of the interpolated image from the downsampled counterpart into consideration. They proposed an interpolation-dependent image downscaling algorithm by modeling the downscaling operation as the inverse operator of upsampling. Benefiting from the well established deep learning frameworks, Hou *et al.* [20] proposed a deep feature consistency network that is applicable to image mapping problems. One of the applications illustrated in the paper is image downscaling. The image downscaling network is trained by keeping the deep features of the input HR image and resulting LR image consistent through another pre-trained deep CNN. Kim *et al.* [21] presented a task aware image downscaling model based on the auto-encoder and the bottleneck layer outputs the downsampled image. In their framework, the encoder acts as the image downscaling network and the decoder is the SR network. The task aware downsampled image is obtained by jointly training the encoder and decoder to maximize the SR performance. Similar to the framework presented by [21], Li *et al.* [4] proposed a convolutional neural network for image compact resolution named CNN-CR, which is composed of a CNN to estimate the LR image and a learned or specified upscaling model to reconstruct the HR image. The generative nature of the encoder like networks implicitly require additional information to constrain the output to be a valid image whose content resembles the HR image. In [20], in order to compute feature consistency loss against the HR image, they upsample the downsampled image back to the same size as the HR input using nearest neighbor interpolation. In [4], [21], guidance images, obtained using

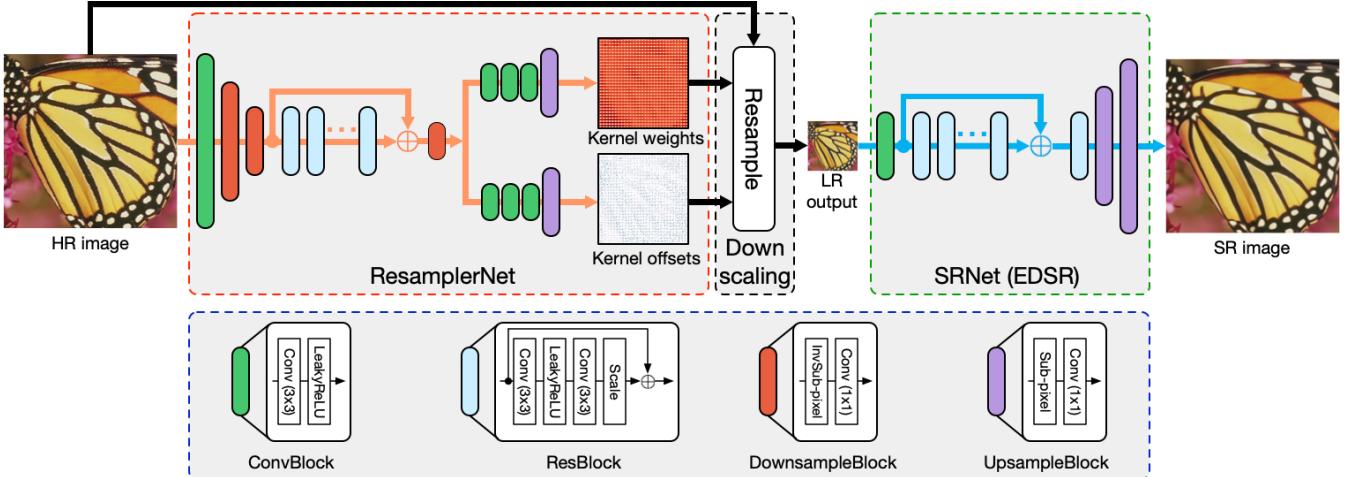


Fig. 1. Network architecture. It consists of three parts, the ResamplerNet, the Downscaling module, and the SRNet. The ResamplerNet is designed to estimate content adaptive resampling kernels and its corresponding offsets for each pixel in the downsampled image. The SRNet, can be any form of differentiable upsampling operations, is employed to guide the training of the ResamplerNet by simply minimizing the SR error. The entire framework is trained end-to-end by back-propagating error signals through the differentiable downscaling module. The composition of each building block is detailed on the blue dashed frame.

bicubic downsampling, are employed to constrain the output space of the LR image generating networks.

### III. MODEL ARCHITECTURE

This section introduces the architecture and formulation of the content adaptive resampler (CAR) model for image downscaling. As shown in Fig. 1, the framework is composed of two major components, *i.e.*, the resampler generation network (ResamplerNet) and the SR network (SRNet). The ResamplerNet is responsible for estimating the content adaptive resampling kernels according to its input HR image, later the resampling kernels are applied to the input HR image to produce the downsampled image. The SRNet takes the resulting downsampled image as input and tries to restore the underlying HR image. The entire framework is trained end-to-end in an unsupervised manner where the primary objective we need to optimize is the SR reconstruction error with respect to the input HR image. By back-propagating the gradient of the reconstruction error through the SRNet and ResamplerNet, the gradient descent algorithm adjusts the parameters of the resampler generation network to produce better resampling kernels which make the downsampled image can be super-resolved more easily.

#### A. Content Adaptive Image Downscaling Resampler

We design the proposed content adaptive image downscaling model that is trained using the unsupervised strategy. Methods presented in [4], [20], [21] synthesize the downsampled image by combining latent representations of the HR image extracted by the CNN and proper constraints are required to make sure that the result is a meaningful image. In this paper, we propose to obtain the downsampled image using the idea of resampling the HR image, which effectively makes the downsampled result look like the original HR image without any constraints.

The filters for traditional bilinear or bicubic downscaling are basically fixed, with the only variation being the shift

of the kernel according to the location of the newly created pixel in the downsampled image. Contrary to this, we propose to use dynamic downscaling filters inspired by the dynamic filter networks [22]. The downscaling kernels are generated for each pixel in the downsampled image depending on the effective resampling region on the HR image. It can be considered as one type of meta-learning [23] that learns how to resample. However, filter-based image resampling methods generally require a certain minimum kernel size to be effective [19]. We alleviate this issue by taking the idea from the deformable convolutional networks [24]. We also associate spatial offset with each element in the resampling kernel. However, different from the deformable convolution operation which parameters are shared spatially, the ResamplerNet estimates different resampling weights for different pixels in the downsampled image. The content adaptive resampling kernels with position offsets can be considered as learnable dilated (atrous) convolutions [25] with the learned dilation rate. Besides, the offset for each kernel element can be different in both magnitude and direction, it can perform non-uniform sampling according to the content structure of the input HR image.

We use a convolutional neural network with residual connections [26] to estimate the weights and offsets for each resampling kernel. The ResamplerNet consists of downscaling blocks, residual blocks and upscaling blocks. The downscaling and residual blocks are trained to model the context of the input HR image as a set of feature maps. Then, two upscaling blocks are used to learn the content adaptive resampling kernel weights  $\mathbf{K} \in \mathbb{R}^{(h/s) \times (w/s) \times m \times n}$ , offsets in the horizontal direction  $\Delta \mathbf{Y} \in \mathbb{R}^{(h/s) \times (w/s) \times m \times n}$  and offsets in the vertical direction  $\Delta \mathbf{X} \in \mathbb{R}^{(h/s) \times (w/s) \times m \times n}$ , respectively.  $h$  and  $w$  are the height and width of the input HR image,  $s$  is the downscaling factor, and  $m, n$  represent the size of the content adaptive resampling kernel. Each kernel is normalized so that elements of a kernel are summed up to be 1.

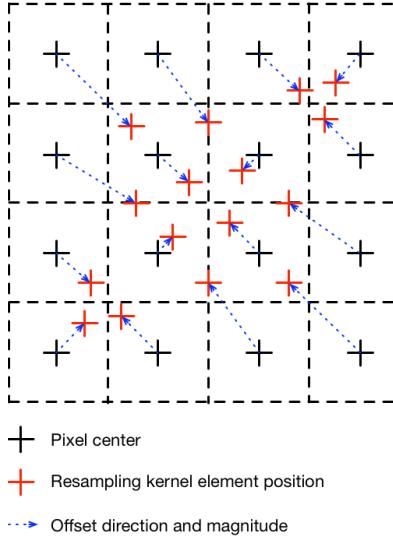


Fig. 2. Non-uniform resampling. The black + represents the center of a pixel in the HR image and the red + indicate the position of the resampling kernel element. The blue dashed arrow shows the offset direction and magnitude of the resampling kernel element relative to its corresponding pixel center.

### B. Image Downscaling

The estimated content adaptive resampling kernels are applied to the corresponding positions of the input HR image to construct the pixel in the downsampled image. For each output pixel, the same resampling kernel is simultaneously applied to three channels of the RGB image. As illustrated in Fig. 2, pixels covered by the resampling kernel are weights summed to obtain the pixel value in the downsampled image.

1) *Forward Pass:* Firstly, we need to position each resampling kernel, associated with the pixel of the downsampled image, on the HR image. It can be achieved using the projection operation defined as:

$$(u, v) = (x + 0.5, y + 0.5) \times scale - 0.5 \quad (1)$$

where  $(x, y)$  is the indices of a pixel at the  $x$ -th row and  $y$ -th column in the downsampled image,  $scale$  represents the downscaling factor, and the resulting  $(u, v)$  is the center of downsampled pixel  $\mathbf{p}_{x,y}^{\text{LR}}$  projected on the input HR image. Equation 1 assumes that pixels have a nonzero size, and the distance between two neighboring samples is one pixel.

Then, each pixel in the downsampled image is created by local filtering on pixels in the input HR image with the corresponding content adaptive resampling kernel as follows:

$$\mathbf{p}_{x,y}^{\text{LR}} = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \mathbf{K}_{x,y}(i, j) \cdot s^{\text{HR}} \left( u + i - \frac{m}{2} + \Delta \mathbf{X}_{x,y}(i, j), v + j - \frac{n}{2} + \Delta \mathbf{Y}_{x,y}(i, j) \right) \quad (2)$$

where  $\mathbf{K}_{x,y} \in \mathbb{R}^{m \times n}$  is the resampling kernel associated with the downsampled pixel at location  $(x, y)$ ,  $\Delta \mathbf{X}_{x,y} \in \mathbb{R}^{m \times n}$  and  $\Delta \mathbf{Y}_{x,y} \in \mathbb{R}^{m \times n}$  are the spatial offset for each element in the  $\mathbf{K}_{x,y}$ . The  $s^{\text{HR}}$  is the sample point value of the input HR image. Due to the location projection and fractional offsets,  $s^{\text{HR}}$  can refer to non-lattice point on the HR image, therefore,  $s^{\text{HR}}$

is computed by bilinear interpolating the nearby four pixels around the fractional position:

$$s_{u',v'}^{\text{HR}} = (1 - \alpha)(1 - \beta) \cdot \mathbf{p}_{\lfloor u' \rfloor, \lfloor v' \rfloor}^{\text{HR}} + \alpha(1 - \beta) \cdot \mathbf{p}_{\lfloor u' \rfloor, \lfloor v' \rfloor + 1}^{\text{HR}} + (1 - \alpha)\beta \cdot \mathbf{p}_{\lfloor u' \rfloor + 1, \lfloor v' \rfloor}^{\text{HR}} + \alpha\beta \cdot \mathbf{p}_{\lfloor u' \rfloor + 1, \lfloor v' \rfloor + 1}^{\text{HR}} \quad (3)$$

where  $u'$  and  $v'$  are fractional position on the HR image,  $\alpha = u' - \lfloor u' \rfloor$  and  $\beta = v' - \lfloor v' \rfloor$  are the bilinear interpolation weights.

2) *Backward Pass:* The ResamplerNet is trained using the gradient descent technique and we need to back-propagate gradients from the SRNet through the resampling operation. The partial derivative of the downsampled pixel with respect to the resampling kernel weight can be formulated as:

$$\begin{aligned} \frac{\partial \mathbf{p}_{x,y}^{\text{LR}}}{\partial \mathbf{K}_{x,y}(i, j)} &= s^{\text{HR}} \left( u + i - \frac{m}{2} + \Delta \mathbf{X}_{x,y}(i, j), v + j - \frac{n}{2} + \Delta \mathbf{Y}_{x,y}(i, j) \right) \end{aligned} \quad (4)$$

the partial derivative of the downsampled pixel with respect to the element in the resampling kernel is simply the interpolated pixel value. Equation 4 is derived with a single channel image and can be generalized to the color image by summing up values calculated separately on the R, G and B channels.

The partial derivative of the downsampled pixel with respect to the kernel element offset is computed as:

$$\begin{aligned} \frac{\partial \mathbf{p}_{x,y}^{\text{LR}}}{\partial \Delta \mathbf{X}_{x,y}(i, j)} &= \frac{\partial \mathbf{p}_{x,y}^{\text{LR}}}{\partial s_{u',v'}^{\text{HR}}} \cdot \frac{\partial s_{u',v'}^{\text{HR}}}{\partial \Delta \mathbf{X}_{x,y}(i, j)} \\ \frac{\partial \mathbf{p}_{x,y}^{\text{LR}}}{\partial \Delta \mathbf{Y}_{x,y}(i, j)} &= \frac{\partial \mathbf{p}_{x,y}^{\text{LR}}}{\partial s_{u',v'}^{\text{HR}}} \cdot \frac{\partial s_{u',v'}^{\text{HR}}}{\partial \Delta \mathbf{Y}_{x,y}(i, j)} \end{aligned} \quad (5)$$

because we employ bilinear interpolation to compute  $s_{u',v'}^{\text{HR}}$ , therefore, the partial derivative of downsampled pixel with respect to the kernel offset is defined as:

$$\begin{aligned} \frac{\partial \mathbf{p}_{x,y}^{\text{LR}}}{\partial \Delta \mathbf{X}_{x,y}(i, j)} &= \mathbf{K}_{x,y}(i, j) \cdot (1 - \beta) \cdot (\mathbf{p}_{\lfloor u' \rfloor, \lfloor v' \rfloor + 1}^{\text{HR}} - \mathbf{p}_{\lfloor u' \rfloor, \lfloor v' \rfloor}^{\text{HR}}) \\ &\quad + \beta \cdot (\mathbf{p}_{\lfloor u' \rfloor + 1, \lfloor v' \rfloor + 1}^{\text{HR}} - \mathbf{p}_{\lfloor u' \rfloor + 1, \lfloor v' \rfloor}^{\text{HR}}) \\ \frac{\partial \mathbf{p}_{x,y}^{\text{LR}}}{\partial \Delta \mathbf{Y}_{x,y}(i, j)} &= \mathbf{K}_{x,y}(i, j) \cdot (1 - \alpha) \cdot (\mathbf{p}_{\lfloor u' \rfloor + 1, \lfloor v' \rfloor}^{\text{HR}} - \mathbf{p}_{\lfloor u' \rfloor, \lfloor v' \rfloor}^{\text{HR}}) \\ &\quad + \alpha \cdot (\mathbf{p}_{\lfloor u' \rfloor + 1, \lfloor v' \rfloor + 1}^{\text{HR}} - \mathbf{p}_{\lfloor u' \rfloor, \lfloor v' \rfloor + 1}^{\text{HR}}) \end{aligned} \quad (6)$$

also because Equation 9 is defined with a single color image, we need to sum up partial derivative of each color component with respect to the offset to obtain the final partial derivative of the downsampled pixel with respect to the kernel element offset.

The pixel value of the downsampled image created using Equation 2 is inherently a continuous floating-point number. However, common image representation describes the color using integer number ranging from 0 to 255, thus, a quantization step is required. Since simply rounding the

floating point number to its nearest integer number is not differentiable, we utilize the soft round method proposed by Nakanishi *et al.* [27] to derive the gradient in the back-propagation during the training phase. The soft round function is defined as:

$$\text{round}_{\text{soft}}(x) = x - \alpha \frac{\sin 2\pi x}{2\pi} \quad (7)$$

where  $\alpha$  is a tuning parameter used to adjust the gradient around the integer position. Note that in the forward propagation, the non-differentiable round function is used to get the nearest integer value.

### C. Image Upscaling

The proposed CAR model is trained using back-propagation to maximize the SR performance. The image upscaling module can be of any form of SR networks, even the differentiable bilinear or bicubic upscaling operations. After the seminal super-resolution model using deep learning proposed by Dong *et al.* [28], *i.e.*, the SRCNN, many state-of-the-art neural SR models have been proposed [29]–[40]. More reviews and discussion about single image SR using deep learning can be referred to [8]. This paper employs state-of-the-art SR model EDSR [33] as the image upscaling module to guide the training of the proposed CAR model.

The EDSR is one of the state-of-the-art deep SR networks and its superior performance is benefited from the powerful residual learning techniques [26]. The EDSR is built on the success of the SRResNet [32] which achieved good performance by simply employing the ResNet [26] architecture on the SR task. The EDSR enhanced SR performance by removing unnecessary parts (Batch Normalization layer [41]) from the SRResNet and also applying a number of tweaks, such as adding residual scaling operation to stable the training process [33]. The SRNet part of Fig. 1 shows the architecture of the EDSR. It is composed of convolution layers converting the RGB image into feature spaces and a group of residual blocks refining feature maps. A global residual connection is employed to improve the efficiency of the gradient back-propagation. Finally, sub-pixel layers are utilized to upsample and transform features into the target SR image.

### D. Training Objectives

One of the main contributions of our work is that we propose a model to learn image downscaling without any supervision signifying that no constraint is applied to the downsampled image. The only objective guiding the generation of the downsampled image is the SR restoration error. The most common loss function generally defaults to the SR network training is the mean squared error (MSE) or the L2 norm loss, but it tends to lead to poor image quality as perceived by human observers [42]. Lim *et al.* [33] found that using another local metric, *e.g.*, L1 norm, can speed up the training process and produce better visual results. In order to improve the visual fidelity of the super-resolved image, perceptually-motivated metrics, such as SSIM [16], MS-SSIM [43] and perceptual loss [44] are usually incorporated in the SR network training. To do fair comparisons with the EDSR, we only adopt the

L1 norm loss as the restoration metric as suggested by [33]. The L1 norm loss defined for SR is:

$$\mathcal{L}^{\text{L1}}(\hat{\mathbf{I}}) = \frac{1}{N} \sum_{p \in \mathbf{I}} |p - \hat{p}| \quad (8)$$

where  $\hat{\mathbf{I}}$  is the SR result,  $p$  and  $\hat{p}$  represent the ground-truth and reconstructed pixel value,  $N$  indicates the number of pixels times the number of color channels.

We associate spatial offset for each element in the resampling kernel, and the offset is estimated without taking the neighborhoods of the kernel element into account. Independent kernel element offset may break the topology of the resampling kernel. To alleviate this problem, we suggest using the total offset distance of all kernel elements as a regularization which encourages kernel elements to stay in their rest position (avoid unnecessary movements). Additionally, since the pixels indexed by the kernel elements that are far from the central position may have less correlation to the resampling result, we assign different weights to their corresponding offset distance in terms of their position relative to the central position. The offset distance regularization term for a single resampling kernel is thus formulated as:

$$\mathcal{L}_{\text{offset}} = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \eta + \sqrt{\Delta \mathbf{X}_{x,y}(i, j)^2 + \Delta \mathbf{Y}_{x,y}(i, j)^2} \cdot w(i, j) \quad (9)$$

where  $w(i, j) = \sqrt{(i - \frac{m}{2})^2 + (j - \frac{n}{2})^2} / \sqrt{\frac{m^2}{2} + \frac{n^2}{2}}$  is the normalized distance weight, the  $\eta$  is introduced to act as the offset distance weight regulator.

The inconsistent resampling kernel offset of spatially neighboring resampling kernels may cause pixel phase shift on the resulting LR images, which is manifested as jaggies, especially on the vertical and horizontal sharp edges (*e.g.*, the LR image in Fig. 5 (b)). To alleviate this phenomenon, we introduce the total variation (TV) loss [45] to constrain the movement of spatially neighboring resampling kernels. Instead of constraining the offsets on both vertical and horizontal directions, we only regularize vertical offsets on the horizontal direction and horizontal offsets on the vertical direction, which we name it the partial TV loss. Besides, variations of each resampling kernel offsets are weighted by its corresponding resampling kernel weights, leading to the following formula:

$$\begin{aligned} \mathcal{L}^{\text{TV}} = & \sum_{x,y} \left( \sum_{i,j} |\Delta \mathbf{X}_{x,y+1}(i, j) - \Delta \mathbf{X}_{x,y}(i, j)| \cdot \mathbf{K}(i, j) \right. \\ & \left. + \sum_{i,j} |\Delta \mathbf{Y}_{x+1,y}(i, j) - \Delta \mathbf{Y}_{x,y}(i, j)| \cdot \mathbf{K}(i, j) \right) \end{aligned} \quad (10)$$

Finally, the optimization objective of the entire framework is defined as:

$$\mathcal{L} = \mathcal{L}^{\text{L1}} + \lambda \overline{\mathcal{L}^{\text{offset}}} + \gamma \mathcal{L}^{\text{TV}} \quad (11)$$

where the  $\overline{\mathcal{L}^{\text{offset}}}$  is the mean offset distance regularization term of all the resampling kernels, and  $\lambda$  is a scalar introduced to control the strength of offset distance regularization.  $\gamma$  is also a scalar used to tune the contribution of the partial TV loss to the final optimization objective.

## IV. EXPERIMENTS

### A. Experimental Setup

1) *Datasets and Metrics:* For training the proposed content adaptive image downscaling resampler generation network under the guidance of EDSR, we employed the widely used DIV2K [46] image dataset. There are 1000 high-quality images in the DIV2K dataset, where 800 images for training, 100 images for validation and the other 100 images for testing. In the testing, four standard datasets, *i.e.*, the Set5 [47], Set14 [48], BSD100 [49] and Urban100 [50] were used as suggested by the EDSR paper [33]. Since we focus on how to downscale images without any supervision, only HR images of the mentioned datasets were utilized. Following the setting in [33], we evaluated the peak noise-signal ratio (PSNR) and SSIM [16] on the Y channel of images represented in the YCbCr (Y, Cb, Cr) color space.

2) *Implementation Details:* Regarding the implementation of the ResamplerNet, we first subtracted the mean RGB value of the DIV2K training set. During the downsampling process, we gradually increased the channels of the output feature map from 3 to 128 using  $3 \times 3$  convolution operation followed by the LeakyReLU activation. 5 residual blocks with each having features of 128 channels are used to model the context. For the two branches estimating the resampling kernels and offsets, we used the same architecture which is composed of ‘Conv-LeakyReLU’ pairs with 256 feature channels. A sub-pixel convolution was applied to upscale and transform the input feature maps into resampling kernels and offsets. For the configuration of the EDSR, we adopted the one with 32 residual blocks and 256 features for each convolution in the residual block.

One of the important hyper-parameters that must be determined is the resampling kernel size and the unit offset length. We defined a  $3 \times 3$  kernel size on the downsampled image space. Its actual size on the HR image space is  $(3 \times s) \times (3 \times s)$ , where  $s$  is the downscaling factor. The unit offset length was defined as one pixel on the downsampled image space whose corresponding unit length on the HR image space is  $s$ . For the offset distance weight regulator in Equation 9, we empirically set it to be 1.

The entire framework was trained on the DIV2K training set using the Adam optimizer [51] with  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$  and  $\epsilon = 10^{-6}$ . We set the mini-batch size as 16, and randomly crop the input HR image into  $192 \times 192$  (for  $4 \times$  downscale and SR) and  $96 \times 96$  (for  $2 \times$  downscale and SR) patches. Training samples were augmented by applying random horizontal and vertical flip. During training, we conducted the validation using 10 images from the DIV2K validation set to select the trained model parameters, and the PSNR on validation was performed on full RGB channels [33]. The initial learning rate was  $10^{-4}$  and decreased when the validation performance does not increase within 100 epoch.

### B. Evaluation of Downscaling Methods for SR

This section reports the quantitative and qualitative performance of different image downscaling methods for SR. Then results of ablation studies of the proposed CAR model

is presented. We compared the CAR model with four image downscaling methods, *i.e.*, the bicubic downscaling (Bicubic), and other three state-of-the-art image downscaling methods: perceptually optimized image downscaling (Perceptually) [15], detail-preserving image downscaling (DPID) [17], and L0-regularized image downscaling (L0-regularized) [19]. We trained SR models using LR images downsampled by those four downscaling algorithms and LR images downsampled by the proposed CAR model. The DPID requires to manually tune a hyper-parameter, which is content variant, to produce better perceptually favorable results. However, it is unpractical for us to generate large amount LR images by manually tuning, also different people may have different perceptual preference. Thus, default value provided by the source code was adopted.

1) *Quantitative and Qualitative Analysis:* Table I summarizes the quantitative comparison results of different image downscaling methods for SR. It consists of two parts, one for bicubic upscaling and one for upscaling using the EDSR. We first analyze the SR performance using the EDSR. As shown in Table I (Upscaling→EDSR), the proposed CAR model trained under the guidance of EDSR considerably boosts the PSNR metric over all the testing cases, and a noticeable gain on the SSIM metric is also obtained. The significant performance gain is benefited from the joint training of the CAR image downscaling model and the EDSR in the end-to-end manner, where the goal of maximizing SR performance encourages the CAR to estimate better resamplers that produce the most suitable downsampled image for SR.

When compared to the SR performance of the LR images downsampled by the bicubic interpolation, the three state-of-the-art image downscaling algorithms can hardly achieve satisfying results, although the visual quality superiority of the downsampled images is reported by those original work. This is because those image downscaling methods are designed for better human perception thus the original information is changed considerably, which makes the downsampled image not well adapted to the SR defined by distortion metrics. Additionally, compared to the SR baseline of the bicubic image downscaling, we note a significant performance degradation on the perceptually based image downscaling method. This indicates that the downsampled image produced by the SSIM optimization cannot be well super-resolved by the state-of-the-art EDSR. The key reason can be illustrated as the SSIM optimization depends on patch selection which may lead to sub-pixel offset in the downsampled image. Other artifacts may underperform SR includes color splitting and noise exaggeration incurred during SSIM optimization [19].

In addition, we also evaluated SR performance of the CAR model trained under the guidance of the bicubic interpolation based upscaling where the bicubic downscaling was used as the baseline. As reported in Table I (Upscaling→Bicubic), the CAR model outperforms the fixed bicubic downscaling methods in terms of upscaling using the fixed bicubic interpolation. The comparison results demonstrate the effectiveness of the proposed CAR model that it is flexible and can be trained under the guidance of differentiable upscaling operations, even if the upscaling operator is not learnable. With this discovery, the proposed CAR model can potentially replace the traditional

TABLE I

QUANTITATIVE EVALUATION RESULTS (PSNR / SSIM) OF DIFFERENT IMAGE DOWNSCALING METHODS FOR SR ON BENCHMARK DATASETS: SET5, SET14, BSD100, URBAN100 AND DIV2K (VALIDATION SET)

Upscaling		Bicubic		EDSR				
Downscaling		Bicubic	CAR	Bicubic	Perceptually	DPIID	L0-regularized	CAR
Set5	2x	33.66 / 0.9299	34.38 / 0.9426	38.06 / 0.9615	31.45 / 0.9212	37.02 / 0.9573	35.40 / 0.9514	38.94 / 0.9658
	4x	28.42 / 0.8104	28.93 / 0.8335	32.35 / 0.8981	25.65 / 0.7805	31.75 / 0.8913	31.05 / 0.8847	33.88 / 0.9174
Set14	2x	30.24 / 0.8688	31.01 / 0.8908	33.88 / 0.9202	29.26 / 0.8632	32.82 / 0.9119	31.56 / 0.9008	35.61 / 0.9404
	4x	26.00 / 0.7027	26.39 / 0.7326	28.64 / 0.7885	24.21 / 0.6684	28.27 / 0.7784	27.67 / 0.7702	30.31 / 0.8382
B100	2x	29.56 / 0.8431	30.18 / 0.8714	32.31 / 0.9021	28.62 / 0.8383	31.47 / 0.8922	30.75 / 0.8816	33.83 / 0.9262
	4x	25.96 / 0.6675	26.17 / 0.6963	27.71 / 0.7432	24.61 / 0.6391	27.27 / 0.7341	27.00 / 0.7293	29.15 / 0.8001
Urban100	2x	26.88 / 0.8403	27.38 / 0.8620	32.92 / 0.9359	26.39 / 0.8483	31.64 / 0.9271	30.23 / 0.9172	35.24 / 0.9572
	4x	23.14 / 0.6577	23.35 / 0.6844	26.62 / 0.8041	21.58 / 0.6295	26.07 / 0.7967	25.83 / 0.7957	29.28 / 0.8711
DIV2K (validation)	2x	31.01 / 0.9393	33.18 / 0.9317	36.76 / 0.9482	31.23 / 0.8984	35.75 / 0.9419	34.69 / 0.9354	38.26 / 0.9599
	4x	26.66 / 0.8521	28.50 / 0.8557	31.04 / 0.8452	26.28 / 0.7381	30.53 / 0.8373	30.18 / 0.8340	32.82 / 0.8837

Note: Red color indicates the best performance and Blue color represents the second.

TABLE II

EVALUATION RESULTS (PSNR / SSIM) OF 4× UPSCALING USING DIFFERENT SR NETWORKS ON BENCHMARK IMAGES DOWNSCALED BY THE CAR MODEL

Upscaling	Downscaling	Set5	Set14	B100	Urban100	DIV2K
SRDenseNet	Bicubic	32.02 / 0.8934	28.50 / 0.7782	27.53 / 0.7337	26.05 / 0.7819	- / -
	CAR†	33.16 / 0.9067	29.85 / 0.8201	28.73 / 0.7794	27.97 / 0.8403	32.24 / 0.8674
	CAR‡	32.63 / 0.9047	29.24 / 0.8122	28.44 / 0.7781	27.12 / 0.8248	31.77 / 0.8654
D-DBPN	Bicubic	32.47 / 0.8980	28.82 / 0.7860	27.72 / 0.7400	26.38 / 0.7946	- / -
	CAR†	33.07 / 0.9061	29.75 / 0.8189	28.70 / 0.7789	27.98 / 0.8376	32.13 / 0.8664
	CAR‡	32.71 / 0.9055	29.17 / 0.8076	28.45 / 0.7784	27.00 / 0.8222	31.76 / 0.8650
RDN	Bicubic	32.47 / 0.8990	28.81 / 0.7871	27.72 / 0.7419	26.61 / 0.8028	- / -
	CAR†	33.34 / 0.9132	29.93 / 0.8308	28.89 / 0.7961	28.53 / 0.8582	32.32 / 0.8756
	CAR‡	33.15 / 0.9112	29.59 / 0.8227	28.79 / 0.7913	27.69 / 0.8412	32.20 / 0.8747
RCAN	Bicubic	32.63 / 0.9002	28.87 / 0.7889	27.77 / 0.7436	26.82 / 0.8087	30.77 / 0.8459
	CAR†	33.84 / 0.9187	30.27 / 0.8383	29.16 / 0.8021	29.23 / 0.8719	32.81 / 0.8842
	CAR‡	33.37 / 0.9138	29.87 / 0.8294	28.95 / 0.7953	28.28 / 0.8541	32.46 / 0.8786

CAR†: the CAR model is trained jointly with its corresponding SR model.

CAR‡: the SR model is trained using the downscaled images generated by the CAR model that is jointly with the EDSR.

Note: Red color indicates the best performance and Blue color represents the second. The ‘-’ indicates that results are not provided by the corresponding original publication.

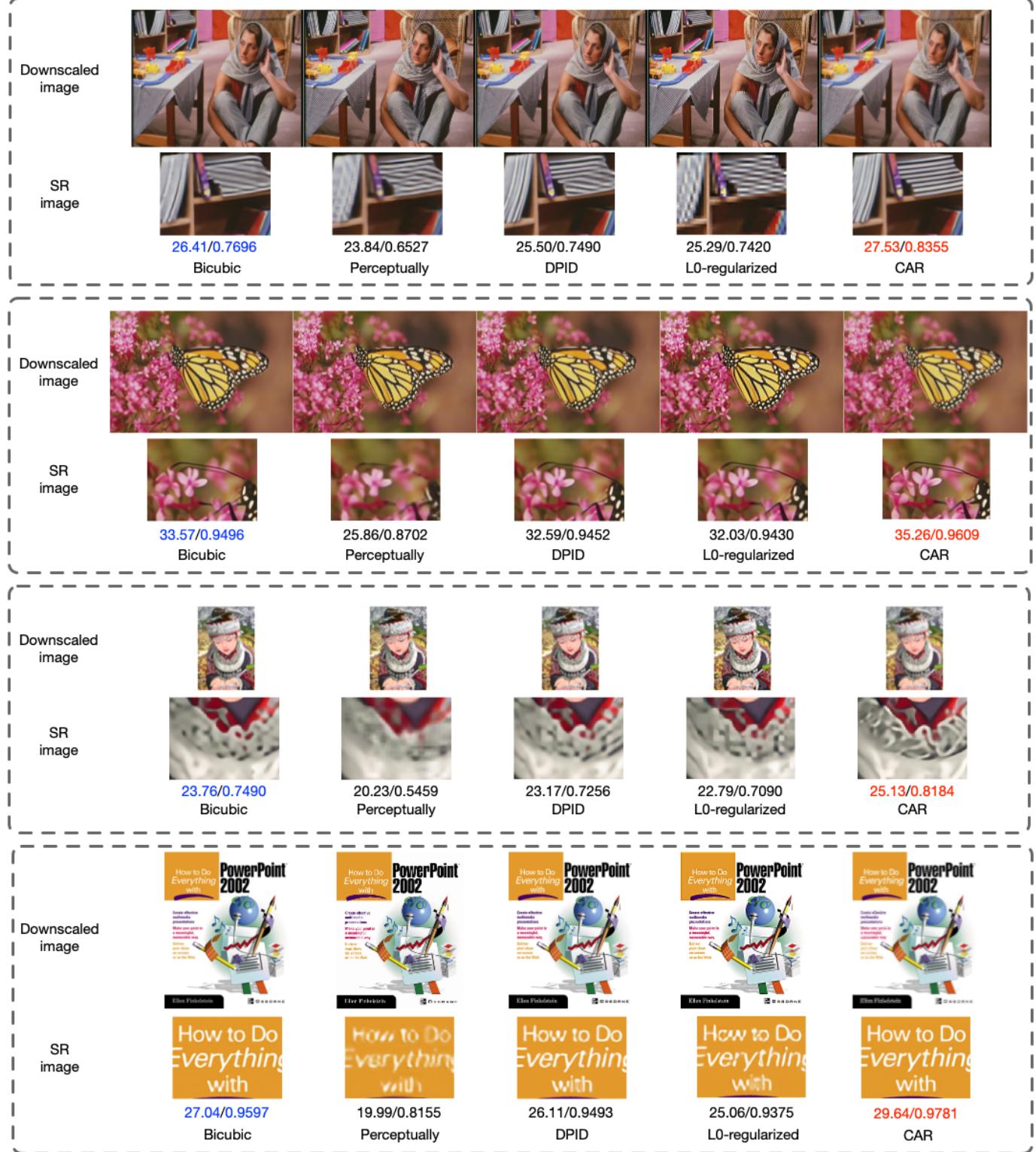
and commonly used bicubic image downscaling operation under the hood, and end-users can obtain extra image zoom in quality gain freely when using the bicubic interpolation for upscaling.

To further validate the effectiveness of the CAR image downscaling model, we evaluated the CAR model trained with another four state-of-the-art deep SR models, *i.e.*, the SRDenseNet [34], D-DBPN [52], RDN [35] and RCAN [37], using 4× downscaling and upscaling factor on five testing datasets. We trained all models using the DIV2K training dataset and all other training setup is set to be the same as described in Section IV-A.2. Table II presents the PSNR and SSIM of 4× upscaled images corresponding to LR images generated using the bicubic interpolation (MATLAB’s `imresize` function with default settings) and the CAR model. Experimental results (Bicubic and CAR†) demonstrate a consistent performance gain of the SR task on images downscaled using the CAR model against that of the bicubic interpolation downscaling. When considering the SR performance of the CAR model trained under the guidance of the bicubic interpolation (Table I) we can reasonably arrive at

the conclusion that the CAR image downscaling model can be learned to adapt to SR models as long as the SR operation is differentiable.

In order to illustrate that the CAR image downscaling model can effectively preserve essential information which can help SR models learn to better recover the original image content, we conducted another experiment. The four state-of-the-art SR models are trained using LR images generated by the proposed CAR model trained jointly with the EDSR. Experimental results shown by the ‘CAR‡’ in Table II indicate that the performance of SR models trained using LR images generated by the CAR model trained jointly with the EDSR significantly surpasses that trained using images downscaled by the bicubic interpolation. We also observed that the performance gain of the CAR‡ against the Bicubic is larger than the performance degradation against the CAR†. The two findings lead to the conclusion that the CAR image downscaling model does preserve content adaptive information that is essential to superior SR using deep SR models.

A qualitative comparison of 4× downscaled images for SR is presented in Fig. 3. As can be seen, the CAR model



This figure is best viewed in color. Zoom in to see details of the downsampled image. The resolution ratio between the downsampled image and SR image is 1. The metric presented under the image is PSNR/SSIM.

Fig. 3. Qualitative results of 4x downsampled image and SR using the EDSR on four example images from the Set14 dataset. (More results are presented in the supplementary file.)

produces downsampled images that are super-resolved with the best visual quality when compared with that of the other four models trained using the EDSR. As shown by the ‘Barbara’ example, due to obvious aliasing occurred during downscaling

by the other four methods, the EDSR cannot recover the correct direction of the parallel edge pattern formed by a stack of books. The downsampled image generated by the CAR incurs less aliasing and the EDSR well recovered the direction of

TABLE III  
ABLATION RESULTS (PSNR / SSIM) OF THE CAR MODEL ON THE SET5, SET14, BSD100, URBAN100 AND DIV2K (VALIDATION SET)

Model	Scale	Set5	Set14	B100	Urban100	DIV2K
CAR	2x	38.94 / 0.9658	35.61 / 0.9404	33.83 / 0.9262	35.24 / 0.9572	38.26 / 0.9599
CAR (w/o TV loss)	2x	39.00 / 0.9663	35.65 / 0.9411	33.91 / 0.9273	35.45 / 0.9587	38.34 / 0.9601
CAR (w/o offset constrain)	2x	38.73 / 0.9641	35.29 / 0.9372	33.58 / 0.9213	35.19 / 0.9560	38.03 / 0.9581
CAR (w/o offset)	2x	38.09 / 0.9600	34.46 / 0.9331	33.25 / 0.9209	32.91 / 0.9423	37.24 / 0.9497
CAR	4x	33.88 / 0.9174	30.31 / 0.8382	29.15 / 0.8001	29.28 / 0.8711	32.82 / 0.8837
CAR (w/o TV loss)	4x	34.13 / 0.9222	30.46 / 0.8439	29.36 / 0.8096	29.36 / 0.8772	33.05 / 0.8893
CAR (w/o offset constrain)	4x	33.86 / 0.9174	30.18 / 0.8368	29.11 / 0.7998	29.02 / 0.8704	32.74 / 0.8835
CAR (w/o offset)	4x	33.11 / 0.9168	29.68 / 0.8322	28.78 / 0.7921	27.98 / 0.8675	32.18 / 0.8830

Note: Red color indicates the best performance and Blue color represents the second.

the parallel edge pattern. For the ‘Comic’ example, we can observe that the SR result of the CAR downscaled image preserves more details. Visual results of the ‘PPT3’ example demonstrates that the SR of downscaled images produced by the CAR better restore continuous edges and produce sharper HR images.

2) *Ablation Studies*: We conducted ablation experiments on the proposed CAR model to verify the effectiveness of our design. We mainly concern about the contribution of the kernel element offset and the constraint on offset distance to the performance of the SR. Table III shows the quantitative ablation results, from which we can observe that the SR performance on all testing cases constantly increases with the addition of kernel element offset and the constraint on offset distance. The baseline model is the CAR without kernel element offset (w/o offset), meaning that the CAR only needs to estimate the resampling kernel weights which will be applied to the position defined by Equation 1 on the HR image (also illustrated as the pixel center in Fig. 2). Then, the kernel element offset was incorporated, which brings a noticeable performance improvement. Introducing kernel element offset makes the resampling kernel to be non-uniform and each element in the resampling kernel can seek to proper sampling position to better preserve useful information for the end SR task. Further SR performance is gained by adding kernel element offset distance regularization. The kernel element offset distance regularization encourages the preservation of the resampling kernel topology and avoids unnecessary kernel element movement on the plain region with less structured texture, which potentially makes the training more stable and easier.

In order to better illustrate how the kernel offset distance regularization works, we visualized an example of resampling kernel elements and its corresponding offsets (Fig. 4) in the configuration of with (w/) and without (w/o) the offset distance regularization. We only visualized the central 9 of many kernel elements and offsets for a better demonstration. Fig. 4 (c) and (d) present kernel elements and offsets by the CAR model trained without offset distance regularization. Fig. 4 (e) and (f) show the kernel elements and offsets estimated by the CAR model trained with offset distance regularization. It can be observed that kernel elements estimated by the CAR model trained with offset distance regularization only present obvious movement on the strong edges and textured regions (the wheel ring and handle), and almost hold still at the rather smoothed

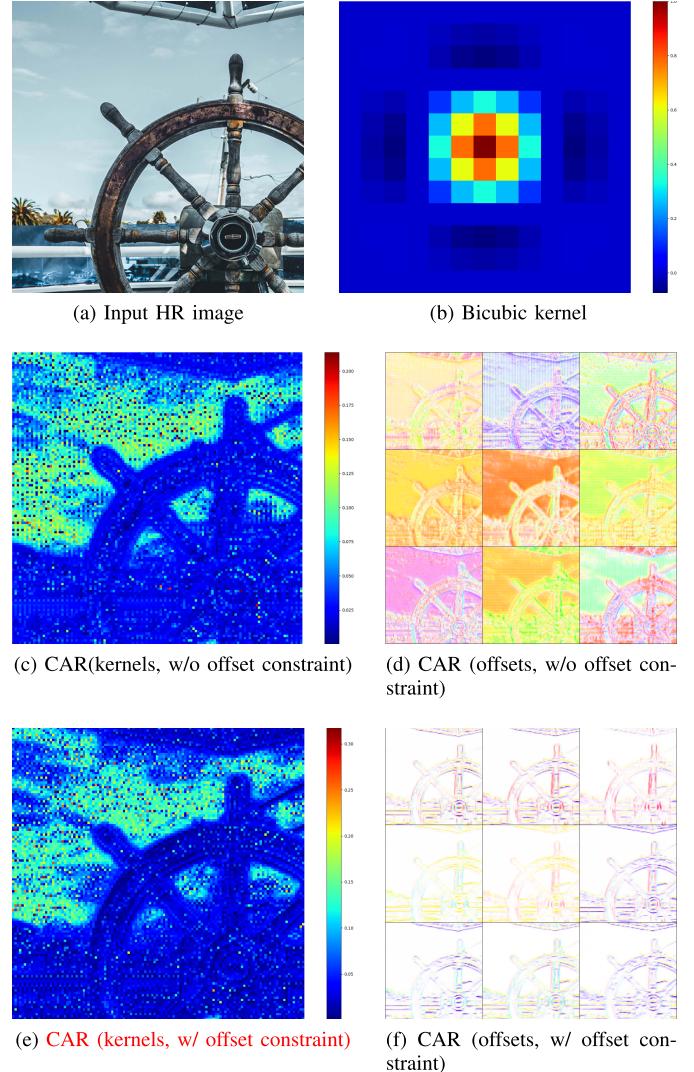


Fig. 4. An example of the 4× downscaled resampling kernel elements and its corresponding offsets. In the figure we only visualized one bicubic resampling kernel, since it will be uniformly applied to all resampling regions. We only visualize the central 9 kernel elements ( $(3 \times h, 3 \times w)$ ) predicted by the CAR model. The kernel element offset is visualized using the color wheel presented in [53].

region (the sky region). The kernel elements estimated by the CAR model trained without offset distance regularization also move towards the strong edges. However, it presents intensive movements on the plain region, which may lead to an unstable training process and sub-optimal testing performance, since the

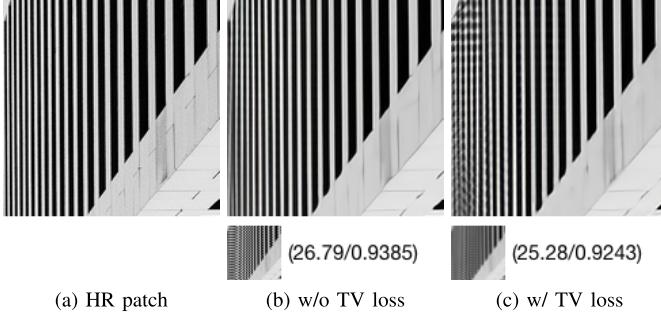


Fig. 5. An example trade off between perception of the  $4\times$  downscaled image and distortion of the  $4\times$  SR image. The first row shows the HR patch (a) and patch of SR results (b and c), the second row shows the downscaled version of the HR patch. The introduction of partial TV loss can produce visually better downscaled image at the expense of some SR performance.

gradient of the resampling kernel depends on the interpolated pixel value (Equation 4). The fixed bicubic kernel is also visualized in Fig. 4 (b). When compared with the content varying kernels shown in Fig. 4 (c) and (e), the fixed bicubic kernel will be uniformly applied to all the resampling region no matter what image content is going to be resampled.

The superior SR performance with the CAR model is achieved from the powerful capability of deep neural networks that can approximate arbitrary functions. However, the deep learning model tends to find a tricky way to produce LR images preserving details that are in favor of generating accurate SR images but not for better human perception. Fig. 5 shows an example of  $4\times$  downscaled image by the CAR and  $4\times$  SR image by the jointly trained EDSR. As shown in Fig. 5 (b), the CAR model learned to preserve more information using much fewer pixel spaces by arranging vertical edges in a regular criss-cross way, which makes the vertical edges in the LR image look jaggy. Jaggies are one type of aliasing that normally manifest as regular artifacts near sharp changes in intensity. However, the human visual system finds regular artifacts more objectionable than irregular artifacts [54]. This problem is possibly caused by the inconsistent movement of the resampling kernels represented by the resampling kernel offsets near the sharp edges. To alleviate it, we introduced the partial TV loss of the horizontal and vertical resampling kernel offsets (Section III-D) to constrain the rather free movements of the resampling kernel elements. As shown in Fig. 5 (c), we can observe a smoother LR image with much less unsightly artifacts.

By employing the partial TV loss of the resampling kernel offsets, the CAR model generates better images for perception. However, we also observed SR performance degradation on each testing dataset, which is shown in the ‘CAR’ and ‘CAR (w/o TV loss)’ entries of Table III. This is because the introduction of the partial TV loss breaks the optimal way of keeping information during downsampling. Other types of aliasing inevitably occurred on the sample-rate conversion, and the SR model cannot recover correct textures from those irregular patterns when compared with that of the regular jaggies. As illustrated by the SR patches shown in Fig. 5 (b) and (c), we can recognize that the SR image corresponding to the

jagged LR image can better represent the original HR patch than that corresponding to the LR image with less jaggies.

*3) Comparison With Public Benchmark SR Results:* To compare the SR performance of downscaled images generated by the proposed CAR model with other deep SR models trained neither on images downscaled using predefined operations (such as bicubic interpolation) or images downscaled by end-to-end trained task driven image downscaling models, we listed several commonly compared public benchmark results on Table IV. The comparison was organized into two groups, SR models trained with LR images produced using the bicubic interpolation method (‘Bicubic’ group) and LR images generated by models trained under the guidance of SR task (‘Learned’ group). In each group, it was further split into models trained using L2 loss function and L1 loss function. SR models in the ‘Bicubic’ group include the SRCNN [28], VDSR [29], DRRN [55], MemNet [56], DnCNN [57], LapSRN [38], ZSSR [58], CARN [59], SRRAM [60]. In the ‘Learned’ group, we compared the CAR model with two recent state-of-the-art image downscaling models trained jointly with deep SR models, *i.e.*, the CNN-CR→CNN-SR [4] model and the TAD→TAU model [21]. We also jointly trained the CAR model with the CNN-SR and TAU to do fair comparisons with the performance reported in its corresponding paper.

As shown in Table IV, SR performance of upscaling models trained jointly with the learnable image downscaling model outperform that of SR models trained using images downscaled in a predefined manner. This is because that LR images generated by learnable image downscaling models adaptively preserve content dependent information during downscaling, which essentially helps deep SR models learn to better recover original image content. Comparing our model with recent state-of-the-art SR driven image downscaling models, the proposed model achieved the best PSNR performance on four out of five testing datasets on the  $2\times$  image downscaling and upscaling track. On the  $4\times$  track, the proposed model achieved the best performance. Noticeable improvement on the Urban100 dataset [50] can be observed. This can be possibly explained by the reason that images of the Urban100 are all buildings with rich sharp edges, and TAD is trained under the constraint of producing LR images similar to images downscaled by bicubic interpolation with pre-filtering which inevitably constrain the edges of LR images generated by the TAD to be blurry like that downscaled by the bicubic interpolation. Therefore, the jointly trained up sampler cannot well recover original edges from those blurred edges of the LR images. As to the proposed CAR model, it is trained without any LR constraint, since the resampling method naturally makes the LR result a valid image. Benefiting from the unsupervised training strategy, the CAR model can adaptively preserve essential edge information for better super-resolution.

### C. Evaluation of Downscaled Images

This section presents the analysis of the quality of the downscaled image produced by the CAR model from two perspectives. We first analyzed the downscaled image in the frequency domain. We used the ‘Barbara’ image from the

TABLE IV

PUBLIC BENCHMARK RESULTS (PSNR / SSIM) OF 2 $\times$  AND 4 $\times$  UPSCALING USING DIFFERENT SR NETWORKS ON THE SET5, SET14, BSD100, URBAN100 AND DIV2K VALIDATION SET

Downscaling type	Loss	Upscaling	Set5	Set14	BSD100	Urban100	DIV2K
Bicubic	L2	SRCNN	36.66 / 0.9542	32.42 / 0.9063	31.36 / 0.8897	29.50 / 0.8946	33.05 / 0.9581
		VDSR	37.53 / 0.9587	33.03 / 0.9213	31.90 / 0.8960	30.76 / 0.9140	33.66 / 0.9625
		DRRN	37.74 / 0.9591	33.23 / 0.9136	32.05 / 0.8973	31.23 / 0.9188	35.63 / 0.9410
		MemNet	37.78 / 0.9597	33.28 / 0.9142	32.08 / 0.8978	31.31 / 0.9195	- / -
		DnCNN	37.58 / 0.9590	33.03 / 0.9118	31.90 / 0.8961	30.74 / 0.9139	- / -
	L1	LapSRN	37.52 / 0.9590	33.08 / 0.9130	31.80 / 0.8950	30.41 / 0.9100	35.31 / 0.9400
		ZSSR	37.37 / 0.9570	33.00 / 0.9108	31.65 / 0.8920	- / -	- / -
		CARN	37.76 / 0.9590	33.52 / 0.9166	32.09 / 0.8978	31.92 / 0.9256	36.04 / 0.9451
		SRRAM	37.82 / 0.9592	33.48 / 0.9171	32.12 / 0.8983	32.05 / 0.9264	- / -
		ESRGAN	- / -	- / -	- / -	- / -	- / -
Learned	L2	(CNN-CR)-(CNN-SR)	38.88 / -	35.40 / -	33.92 / -	33.68 / -	- / -
		(CAR)-(CNN-SR)	38.91 / 0.9656	35.55 / 0.9401	33.96 / 0.9281	34.73 / 0.9539	38.15 / 0.9593
		(CAR)-(EDSR)	39.01 / 0.9662	35.64 / 0.9406	33.84 / 0.9262	35.15 / 0.9568	38.21 / 0.9597
	L1	(TAD)-(TAU)	37.69 / -	33.90 / -	32.62 / -	31.96 / -	36.13 / -
		(CAR)-(TAU)	37.93 / 0.9628	34.19 / 0.9312	32.78 / 0.7592	32.54 / 0.9388	36.86 / 0.9524
		(CAR)-(EDSR)	38.94 / 0.9658	35.61 / 0.9404	33.83 / 0.9262	35.24 / 0.9572	38.26 / 0.9599
Bicubic	L2	SRCNN	30.48 / 0.8628	27.49 / 0.7503	26.90 / 0.7101	24.52 / 0.7221	27.78 / 0.8753
		VDSR	31.35 / 0.8838	28.01 / 0.7674	27.29 / 0.7251	25.18 / 0.7524	28.17 / 0.8841
		SRResNet	32.05 / 0.8910	28.53 / 0.7804	27.57 / 0.7354	26.07 / 0.7839	- / -
		DRRN	31.68 / 0.8888	28.21 / 0.7720	27.38 / 0.7284	25.44 / 0.7638	29.98 / 0.8270
		MemNet	31.74 / 0.8893	28.26 / 0.7723	27.40 / 0.7281	25.50 / 0.7630	- / -
	L1	DnCNN	31.40 / 0.8845	28.04 / 0.7672	27.29 / 0.7253	25.20 / 0.7521	- / -
		LapSRN	31.54 / 0.8850	28.19 / 0.7720	27.32 / 0.7280	25.21 / 0.7560	29.88 / 0.8250
		ZSSR	31.13 / 0.8796	28.01 / 0.7651	27.12 / 0.7211	- / -	- / -
		CARN	32.13 / 0.8937	28.60 / 0.7806	27.58 / 0.7349	26.07 / 0.7837	30.43 / 0.8374
		SRRAM	32.13 / 0.8932	28.54 / 0.7800	27.56 / 0.7350	26.05 / 0.7834	- / -
Learned	L2	ESRGAN	32.73 / 0.9011	28.99 / 0.7917	27.85 / 0.7455	27.03 / 0.8153	- / -
		- / -	- / -	- / -	- / -	- / -	- / -
		(CNN-CR)-(CNN-SR)	33.73 / 0.9148	30.22 / 0.8329	29.13 / 0.7992	28.56 / 0.8546	32.67 / 0.8800
	L1	(CAR)-(CNN-SR)	33.87 / 0.9176	30.34 / 0.8381	29.18 / 0.8005	29.23 / 0.8710	32.85 / 0.8835
		(CAR)-(EDSR)	- / -	- / -	- / -	- / -	- / -
		(TAD)-(TAU)	31.59 / -	28.36 / -	27.57 / -	25.56 / -	30.25 / -
Bicubic	L2	(CAR)-(TAU)	31.85 / 0.8938	28.77 / 0.8028	27.84 / 0.7592	26.15 / 0.7978	31.10 / 0.8512
		(CAR)-(EDSR)	33.88 / 0.9174	30.31 / 0.8382	29.15 / 0.8001	29.28 / 0.8711	32.82 / 0.8837
		- / -	- / -	- / -	- / -	- / -	- / -
		(CNN-CR)-(CNN-SR)	33.87 / 0.9176	30.34 / 0.8381	29.18 / 0.8005	29.23 / 0.8710	32.85 / 0.8835
		(CAR)-(EDSR)	- / -	- / -	- / -	- / -	- / -
	L1	(TAD)-(TAU)	31.59 / -	28.36 / -	27.57 / -	25.56 / -	30.25 / -
		(CAR)-(TAU)	31.85 / 0.8938	28.77 / 0.8028	27.84 / 0.7592	26.15 / 0.7978	31.10 / 0.8512
		(CAR)-(EDSR)	33.88 / 0.9174	30.31 / 0.8382	29.15 / 0.8001	29.28 / 0.8711	32.82 / 0.8837
		- / -	- / -	- / -	- / -	- / -	- / -
		(CNN-CR)-(CNN-SR)	33.87 / 0.9176	30.34 / 0.8381	29.18 / 0.8005	29.23 / 0.8710	32.85 / 0.8835

Note: Red color indicates the best performance and Blue color represents the second. The ‘-’ indicates that results are not provided by the corresponding original publication.

Set14 dataset as an example because it contains a wealth of high-frequency components (as shown in the first example of Fig. 3). Fig. 6 shows the spectrum obtained by applying DFT on the HR image and the downsampled images generated by the CAR model and four downscaling methods been compared. Each point in the spectrum represents a particular frequency contained in the spatial domain image. The point in the center of the spectrum is the DC component, and points closely around the center point represents low-frequency components. The further away from the center a point is, the higher is its corresponding frequency.

The spectrum of the HR image (Fig. 6 (a)) contains a lot of high-frequency components. During image downscaling, spatial-aliasing inevitably occurs since the sampling rate is below the Nyquist frequency. Aliasing can be spotted in the spectrum as spurious bands that are not presented in the spectrum of the HR image (high frequency component is aliased into low frequency), e.g., the black box marked regions in Fig. 6 (c-e) compared to that of the original spectrum in Fig. 6 (a). One way to remove aliasing is to use a blurry filter upon resampling (so does the default MATLAB `imresize` function). As shown by the black box marked regions in Fig. 6 (b), aliasing of the downsampled image produced by the MATLAB `imresize` function is alleviated. Compared

with the spectrum of the image produced by the three state-of-the-art image downscaling methods (Fig. 6 (c-e)), less power exaggeration of low frequency components can be observed in the black box indicated regions. Therefore, the spectrum of the CAR generated image demonstrates that less aliasing is produced during the downscaling by the CAR model. A cropped region from the downsampled ‘Barbara’ image is shown in Fig. 6 (g) and less spatial aliasing can be observed from the Bicubic and CAR downsampled image when compared with that of the other three image downscaling methods.

Additionally, we compared the image quality of the downsampled image from the perspective of lossless compression. In general, compressing high quality image needs more bits-per-pixel (bpp). Table V presents the average bpp of the lossless compressed images using the lossless JPEG (JPEG-LS) [61]. Downsampled images produced by the CAR model can be more easily compressed than that by downscaling algorithms designed for better human perception. The reason why this happened is that downscaling methods designed for better human perception tend to preserve or enhance edges in a pre-defined manner, and much more high frequency components are retained, thus the downsampled images are less compressible on average. Therefore, less perception quality is achieved by the CAR downsampled image when compared with

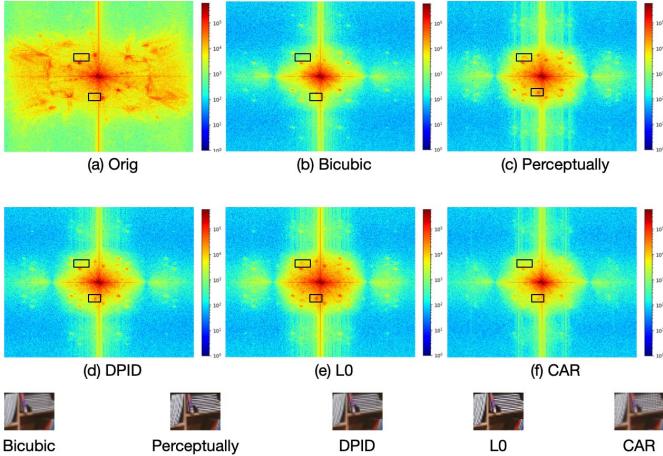


Fig. 6. Spectrum analysis of the  $4 \times$  downsampled ‘Barbara’ image in the Set14 dataset using different downscaling methods. The last row shows the cropped patch from the downsampled ‘Barbara’ image. The resolution ratio between downsampled image patch and the spectrum image is 1.

TABLE V

BPP OF LOSSLESS COMPRESSED DOWNSCALED IMAGES  
USING THE JPEG-LS

		Bicubic	Perceptually	DPID	$L_0$	CAR
Set5		11.99	13.27	12.65	16.10	11.80
Set14		11.23	12.01	12.05	15.07	11.14
BSD100		11.11	11.90	11.8	15.31	10.69
Urban100		11.39	12.22	12.19	15.27	11.28
DIV2K		10.42	11.38	11.1	14.04	10.20
Average		11.228	12.156	11.958	15.158	11.022
Set5	2x	14.78	16.52	15.32	17.41	14.39
Set14	2x	12.61	14.27	13.32	15.19	12.47
BSD100	2x	12.85	14.42	13.51	15.59	12.61
Urban100	2x	12.23	14.26	12.94	14.97	12.28
DIV2K	2x	11.54	13.46	12.14	13.91	11.39
Average	2x	12.802	14.586	13.446	15.414	12.628

Note: Red color indicates the best performance and Blue color represents the second.

that produced by the perception oriented image downscaling methods. When compared with the bpp of compressed bicubic downsampled images, the CAR model achieved similar compression performance, thus, we can infer that the quality of the CAR downsampled image is similar to the bicubic downsampled image.

#### D. User Study

To evaluate the visual quality of the generated SR images and LR images corresponding to different downscaling algorithms, we conducted user study which is widely adopted in many image generation tasks. We picked 59 sample images from different testing datasets, *i.e.*, the Set5 (5 images), Set14 (20 images), BSD100 (20 images), and Urban100 (20 images) dataset. Samples are randomly selected with diverse properties, including people, animals, buildings, natural scenes, and computer-generated graphics. We adopted similar evaluation settings used in [14], [15], [17], [19]. The user study was conducted as the A/B testing: the original image was presented in the middle place with two variants (SR images or downsampled images) showed in either side, among which one is produced by the CAR method and another is generated by

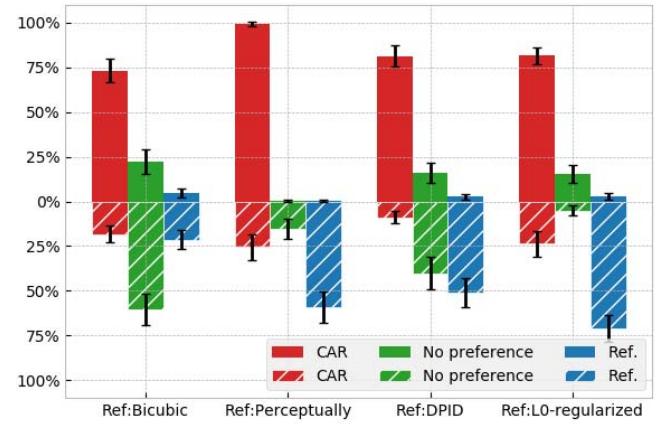


Fig. 7. User study results comparing the proposed CAR model against reference image downscaling algorithms for the image super-resolution and image downscaling tasks. Each data point is an average over valid records evaluated on 20 image groups and the error bars indicate 95% confidence interval. The upper part (above the zero axes) is the image quality preference of the SR task, and the lower part (below the zero axes) is the image quality preference of the image downscaling task.

one of the competing methods, *i.e.*, Bicubic, Perceptually [15], DPID [17] and L0-regularized [19]. Users were required to answer the question ‘which one looks better’ by exclusively selecting one of the three options from 1) A is better than B; 2) A equals to B; 3) B is better than A. For all 59 sample images, there are 236 pairwise decisions for each user. All image pairs were shown in random temporal order and the two variants of the original image of a pair are also randomly shown in position A or position B. To test the reliability of the user study result, we add additional 10 image groups by randomly repeating question from the 236 trials. All images were displayed at the native resolution of the monitor and the zoom functionality of the UI was disabled. Users can only pan the view if the total size of a group of images exceeds the screen resolution. No other restrictions on the viewing way were imposed and users can judge those images at any viewing distance and angle without time limits.

We invited 29 participants for both the user study on  $4 \times$  image super-resolution and  $4 \times$  image downscaling tasks. Answers from each participant are filtered out if it achieves less than 80% consistency [14], [15] on the repeated 10 questions, which generates 29 and 28 valid records for the image super-resolution and image downscaling tasks, respectively. As can be seen from Fig. 7, the total preferences of the SR and image downscaling task opted for the CAR model are larger than that of the reference methods. The upper part (above the zero axes) of Fig. 7 shows the results of the user study on  $4 \times$  image super-resolution task, each group of bars presents the comparison between SR images corresponding to the CAR generated LR images and SR images corresponding to LR images produced by the reference method. The results indicate that the CAR model achieves at least 73% preference over all other algorithms. Besides, our algorithm achieves more than 98% preference compared with the Perceptually based [15] downsampled image and the original HR image. Although there

are about 20% preference on ‘A equals to B’ plus ‘B is better than A’ on the DPID and L0-regularized entry, it still cannot compete with the significant superiority of the CAR method. When combined with the objective metrics presented in Table I, we can arrive at the conclusion that LR images produced by algorithms solely optimized for better human perception cannot be well recovered.

The lower part (below the zero axes) of Fig. 7 shows the users’ perceptual preference for the  $4 \times$  downsampled images generated by the CAR method versus the Bicubic, Perceptually, DPID, and L0-regularized image downscaling algorithms. We observed that the CAR method gets distinct less preference when compared with that of the Perceptually, DPID, and L0-regularized algorithm, which illustrates that the three state-of-the-art algorithms generate more perceptually favored LR images. However, when compared with the Bicubic downscaling algorithm, the user preference for the CAR method is slightly inferior to that for the Bicubic, and in most cases, participants tend to give no preference for both methods. This indicates that the CAR image downscaling method is comparable to the bicubic downscaling algorithm in terms of human perception. Among the three state-of-the-art image downscaling algorithms, the DPID achieves less agreement since its hyper-parameter is content dependent, and during the test, the default value is used. The perceptually based image downscaling and the L0-regularized method achieve more than 75% user preference because these methods artificially emphasize the edges of image content, which can be good if the image is going to be displayed at a very small size, such as an icon. Whether it is desirable in other cases is debatable. It does tend to make images look better at first glance, but at the expense of realism in terms of signal fidelity.

## V. CONCLUSION

In this paper, we introduce the CAR model for image downscaling, which is an end-to-end system trained by maximizing the SR performance. It simultaneously learns a mapping for resolution reduction and SR performance improvement. One major contribution of our work is that the CAR model is trained in an unsupervised manner meaning that there is no assumption on how the original HR image will be downsampled, which helps the image downscale model to learn to keep essential information for SR task in a more optimal way. This is achieved by the content adaptive resampling kernel generation network which estimates spatial non-uniform resampling kernels for each pixel in the downsampled image according to the input HR image. The downsampled pixel value is obtained by decimating HR pixels covered by the resampling kernel. Our experimental results illustrate that the CAR model trained jointly with the SR networks achieves a new state-of-the-art SR performance while produces downsampled images whose quality is comparable to that of the widely adopted image downscaling method.

## REFERENCES

- [1] A. M. Bruckstein, M. Elad, and R. Kimmel, “Down-scaling for better transform compression,” *IEEE Trans. Image Process.*, vol. 12, no. 9, pp. 1132–1144, Sep. 2003.
- [2] W. Lin and L. Dong, “Adaptive downsampling to improve image compression at low bit rates,” *IEEE Trans. Image Process.*, vol. 15, no. 9, pp. 2513–2521, Sep. 2006.
- [3] Y. Zhang, D. Zhao, J. Zhang, R. Xiong, and W. Gao, “Interpolation-dependent image downsampling,” *IEEE Trans. Image Process.*, vol. 20, no. 11, pp. 3291–3296, Nov. 2011.
- [4] Y. Li, D. Liu, H. Li, L. Li, Z. Li, and F. Wu, “Learning a convolutional neural network for image compact-resolution,” *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1092–1107, Mar. 2019.
- [5] C. E. Shannon, “Communication in the presence of noise,” *Proc. IEEE*, vol. 37, no. 1, pp. 10–21, Jan. 1949.
- [6] J. Yang, J. Wright, T. S. Huang, and Y. Ma, “Image super-resolution via sparse representation,” *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2873, Nov. 2010.
- [7] J. Yang and T. Huang, *Image Super-Resolution: Historical Overview and Future Challenges*. Boca Raton, FL, USA: CRC Press, 2017, pp. 1–33.
- [8] W. Yang, X. Zhang, Y. Tian, W. Wang, J.-H. Xue, and Q. Liao, “Deep learning for single image super-resolution: A brief review,” *IEEE Trans. Multimedia*, vol. 21, no. 12, pp. 3106–3121, Dec. 2019.
- [9] S. Anwar, S. Khan, and N. Barnes, “A deep journey into super-resolution: A survey,” 2019, *arXiv:1904.07523*. [Online]. Available: <https://arxiv.org/abs/1904.07523>
- [10] G. Wolberg, *Digital Image Warping* (IEEE Computer Society Press Monograph). Los Alamitos, CA, USA: IEEE Computer Society Press, 1990.
- [11] L. Fang, O. C. Au, K. Tang, and A. K. Katsaggelos, “Antialiasing filter design for subpixel downsampling via frequency-domain analysis,” *IEEE Trans. Image Process.*, vol. 21, no. 3, pp. 1391–1405, Mar. 2012.
- [12] D. P. Mitchell and A. N. Netravali, “Reconstruction filters in computer-graphics,” *SIGGRAPH Comput. Graph.*, vol. 22, no. 4, pp. 221–228, Aug. 1988.
- [13] C. E. Duchon, “Lanczos filtering in one and two dimensions,” *J. Appl. Meteor.*, vol. 18, no. 8, pp. 1016–1022, Aug. 1979.
- [14] J. Kopf, A. Shamir, and P. Peers, “Content-adaptive image downscaling,” *ACM Trans. Graph.*, vol. 32, no. 6, pp. 1–8, Nov. 2013.
- [15] A. C. Öztireli and M. Gross, “Perceptually based downscaling of images,” *ACM Trans. Graph.*, vol. 34, no. 4, Jul. 2015, Art. no. 77.
- [16] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
- [17] N. Weber, M. Waechter, S. C. Amend, S. Guthe, and M. Goesele, “Rapid, detail-preserving image downscaling,” *ACM Trans. Graph.*, vol. 35, no. 6, pp. 1–6, Nov. 2016.
- [18] E. S. L. Gastal and M. M. Oliveira, “Spectral remapping for image downscaling,” *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–16, Jul. 2017.
- [19] J. Liu, S. He, and R. W. H. Lau, “ $L_0$ -regularized image downscaling,” *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1076–1085, Mar. 2018.
- [20] X. Hou *et al.*, “Learning based image transformation using convolutional neural networks,” *IEEE Access*, vol. 6, pp. 49779–49792, 2018.
- [21] H. Kim, M. Choi, B. Lim, and K. M. Lee, “Task-aware image downscaling,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 399–414.
- [22] B. De Brabandere, X. Jia, T. Tuytelaars, and L. Van Gool, “Dynamic filter networks,” in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 667–675.
- [23] J. Schmidhuber, “Evolutionary principles in self-referential learning, or on learning how to learn: The meta-meta-hook,” Ph.D. dissertation, Technische Universität München, Munich, Germany, 1987.
- [24] J. Dai *et al.*, “Deformable convolutional networks,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 764–773.
- [25] F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” in *Proc. Int. Conf. Learn. Representations*, 2016, pp. 1–13.
- [26] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [27] K. Nakanishi, S.-I. Maeda, T. Miyato, and D. Okanohara, “Neural multi-scale image compression,” 2019, *arXiv:1805.06386*. [Online]. Available: <https://arxiv.org/abs/1805.06386>
- [28] C. Dong, C. C. Loy, K. He, and X. Tang, “Image super-resolution using deep convolutional networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.
- [29] J. Kim, J. K. Lee, and K. M. Lee, “Accurate image super-resolution using very deep convolutional networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1646–1654.
- [30] J. Kim, J. K. Lee, and K. M. Lee, “Deeply-recursive convolutional network for image super-resolution,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1637–1645.

- [31] W. Shi *et al.*, “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1874–1883.
- [32] C. Ledig *et al.*, “Photo-realistic single image super-resolution using a generative adversarial network,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 105–114.
- [33] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, “Enhanced deep residual networks for single image super-resolution,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1132–1140.
- [34] T. Tong, G. Li, X. Liu, and Q. Gao, “Image super-resolution using dense skip connections,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4809–4817.
- [35] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, “Residual dense network for image super-resolution,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2472–2481.
- [36] K. Zhang, W. Zuo, and L. Zhang, “Learning a single convolutional super-resolution network for multiple degradations,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3262–3271.
- [37] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, “Image super-resolution using very deep residual channel attention networks,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 286–301.
- [38] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, “Fast and accurate image super-resolution with deep Laplacian pyramid networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 11, pp. 2599–2613, Nov. 2019.
- [39] K. Zhang, W. Zuo, and L. Zhang, “Deep plug-and-play super-resolution for arbitrary blur kernels,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1671–1681.
- [40] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang, “Second-order attention network for single image super-resolution,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 11065–11074.
- [41] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proc. 32nd Int. Conf. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.
- [42] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, “Loss functions for image restoration with neural networks,” *IEEE Trans. Comput. Imag.*, vol. 3, no. 1, pp. 47–57, Mar. 2017.
- [43] Z. Wang, E. Simoncelli, and A. Bovik, “Multiscale structural similarity for image quality assessment,” in *Proc. 37th Asilomar Conf. Signals, Syst. Comput.*, Jul. 2004, pp. 1398–1402.
- [44] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 694–711.
- [45] L. I. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” *Phys. D, Nonlinear Phenomena*, vol. 60, nos. 1–4, pp. 259–268, Nov. 1992.
- [46] E. Agustsson and R. Timofte, “NTIRE 2017 challenge on single image super-resolution: Dataset and study,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1122–1131.
- [47] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. A. Morel, “Low-complexity single-image super-resolution based on nonnegative neighbor embedding,” in *Proc. Brit. Mach. Vis. Conf.*, 2012, p. 135.
- [48] R. Zeyde, M. Elad, and M. Protter, “On single image scale-up using sparse-representations,” in *Proc. 7th Int. Conf. Curves Surf.*, 2012, pp. 711–730.
- [49] D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2002, pp. 416–423.
- [50] J.-B. Huang, A. Singh, and N. Ahuja, “Single image super-resolution from transformed self-exemplars,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5197–5206.
- [51] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–15.
- [52] M. Haris, G. Shakhnarovich, and N. Ukita, “Deep back-projection networks for super-resolution,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1664–1673.
- [53] S. Baker, S. Roth, D. Scharstein, M. J. Black, J. P. Lewis, and R. Szeliski, “A database and evaluation methodology for optical flow,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.
- [54] R. L. Cook, “Stochastic sampling in computer graphics,” *TOGACM Trans. Graph.*, vol. 5, no. 1, pp. 51–72, Jan. 1986.
- [55] Y. Tai, J. Yang, and X. Liu, “Image super-resolution via deep recursive residual network,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2790–2798.
- [56] R. Chen, Y. Qu, K. Zeng, J. Guo, C. Li, and Y. Xie, “Persistent memory residual network for single image super resolution,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2018, pp. 922–9227.
- [57] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, “Beyond a Gaussian Denoiser: Residual learning of deep CNN for image denoising,” *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.
- [58] A. Shocher, N. Cohen, and M. Irani, “Zero-shot super-resolution using deep internal learning,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3118–3126.
- [59] N. Ahn, B. Kang, and K.-A. Sohn, “Fast, accurate, and lightweight super-resolution with cascading residual network,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 252–268.
- [60] J.-H. Kim, J.-H. Choi, M. Cheon, and J.-S. Lee, “RAM: Residual attention module for single image super-resolution,” 2018, *arXiv:1811.12043*. [Online]. Available: <https://arxiv.org/abs/1811.12043>
- [61] M. Weinberger, G. Seroussi, and G. Sapiro, “The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS,” *IEEE Trans. Image Process.*, vol. 9, no. 8, pp. 1309–1324, Aug. 2000.



**Wanjie Sun** received the B.Eng. degree in computer science and technology and the M.Eng. degree in information and communication engineering from Nantong University in 2014 and 2017, respectively. He is currently pursuing the Ph.D. degree with the School of Remote Sensing and Information Engineering, Wuhan University. His research interests include computer vision and machine learning.



**Zhenzhong Chen** (Senior Member, IEEE) received the B.Eng. degree from the Huazhong University of Science and Technology and the Ph.D. degree from The Chinese University of Hong Kong, all in electrical engineering. He is currently a Professor with Wuhan University (WHU). His current research interests include image and video processing and analysis, computational vision and human–computer interaction, multimedia data mining, photogrammetry, and remote sensing. He was a recipient of the CUHK Young Scholars Dissertation Award, the CUHK Faculty of Engineering Outstanding Ph.D. Thesis Award, and the Microsoft Fellowship. He has been a VQEG Board Member and the Immersive Media Working Group Co-Chair, a Selection Committee Member of ITU Young Innovators Challenges, an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (TCSVT), the Journal of the Association for Information Science and Technology (JASIST), and the Journal of Visual Communication and Image Representation, and an Editor of the IEEE IoT Newsletters.