SRWarp: Generalized Image Super-Resolution under Arbitrary Transformation

Sanghyun Son Kyoung Mu Lee
ASRI, Department of ECE, Seoul National University, Seoul, Korea
{thstkdgus35, kyoungmu}@snu.ac.kr

Abstract

Deep CNNs have achieved significant successes in image processing and its applications, including single image super-resolution (SR). However, conventional methods still resort to some predetermined integer scaling factors, e.g., $\times 2$ or $\times 4$. Thus, they are difficult to be applied when arbitrary target resolutions are required. Recent approaches extend the scope to real-valued upsampling factors, even with varying aspect ratios to handle the limitation. In this paper, we propose the SRWarp framework to further generalize the SR tasks toward an arbitrary image transformation. We interpret the traditional image warping task, specifically when the input is enlarged, as a spatially-varying SR problem. We also propose several novel formulations, including the adaptive warping layer and multiscale blending, to reconstruct visually favorable results in the transformation process. Compared with previous methods, we do not constrain the SR model on a regular grid but allow numerous possible deformations for flexible and diverse image editing. Extensive experiments and ablation studies justify the necessity and demonstrate the advantage of the proposed SRWarp method under various transformations.

1. Introduction

As one of the fundamental vision problems, image super-resolution (SR) aims to reconstruct a high-resolution (HR) image from a given low-resolution (LR) input. The SR methods are widely used in several applications such as perceptual image enhancement [25, 39], editing [2, 30], and digital zooming [42], due to its practical importance. Similar to the other vision-related tasks, recent convolutional neural networks (CNNs) have achieved promising SR performance with large-scale datasets [1, 27], efficient structures [51, 52, 12], and novel optimization techniques [27, 39]. Recent state-of-the-art methods can reconstruct sharp edges and crisp textures with fine details up to $\times 4$ or $\times 8$ scaling factors on various types of input data including real-world images [50, 4, 41], videos [38, 43, 13, 36, 26], hyperspectral [9, 48, 45], and light field arrays [49, 35, 40].

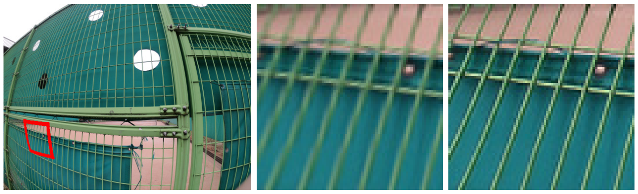

(a) Input (b) OpenCV (c) **SRWarp**

Figure 1: **Real-world lens distortion correction using the proposed SRWarp.** The image is captured by the GoPro HERO6 handheld camera. Our SRWarp implements SR with locally-varying scale factors, which can be used to transform an input image to the desired geometry.

From the perspective of image editing applications, the SR algorithm supports users to effectively increase the number of pixels in the image and reconstruct high-frequency details when its HR counterpart is unavailable. Such manipulation may include simple resizing with some predefined scaling factors or synthesizing images of arbitrary target resolutions. However, directly applying existing SR methods in these situations is difficult because the models are usually designed to cope with some fixed integer scales [25, 51, 39]. Recently, few methods have extended the scope of the SR to upsample a given image by arbitrary scaling factors [15] and aspect ratios [37]. These novel approaches provide more flexibility and versatility to existing SR applications for their practical usage.

Nevertheless, existing SR models are not fully optimized for general image editing tasks due to their intrinsic formulations. Previous approaches are also designed to take and reconstruct such rectangular frames because digital images are defined on a rectangular grid. On the contrary, images may undergo various deformations in practice to effectively adjust their contents within the context. For instance, homographic transformation [47, 24] aligns images from different views, and cameras incorporate various correction algorithms to remove distortions from the lens [34, 44]. One of the shortcomings in the conventional warping methods is that interpolation-based algorithms tend to generate blurry results when a local region of the image is stretched. Hence, an appropriate enhancement algorithm is required to preserve sharp edges and detailed textures, as the SR methods for image upscaling. However, such applications may require images to be processed on irregular grids, which cannot be handled by naïve CNNs for regular-shaped data.

To prevent blurry warping results, state-of-the-art SR models may be introduced prior to image transformation. By doing so, supersampled pixels alleviate blurriness and artifacts from simple interpolation. However, Hu *et al.* [15] and Wang *et al.* [37] have demonstrated that the solution is suboptimal in the arbitrary-scale SR task. Furthermore, a generalized warping algorithm should deal with more complex and even spatially-varying deformations, which are not straightforward to be considered in the previous approaches. Therefore, an appropriate solution is required to effectively combine the SR and warping methods in a single pipeline.

In this study, we interpret the general image transformation task as a spatially-varying SR problem. For such purpose, we construct an end-to-end learnable framework, that is, SRWarp. Different from the previous SR methods, our method is designed to handle the warping problem in two specific aspects. First, we introduce an adaptive warping layer (AWL) to dynamically predict appropriate resampling kernels for different local distortions. Second, our multiscale blending strategy combines features of various resolutions based on their contents and local deformations to utilize richer information from a given image. With powerful backbones [27, 39], the proposed SRWarp can successfully reconstruct image structures that can be missed from conventional warping methods, as shown in Figure 1. Our contributions can be organized in threefolds as follows:

- The novel SRWarp model generalizes the concept of SR under arbitrary transformations and formulates a framework to learn image transformation.
- Extensive analysis shows that our adaptive warping layer and multiscale blending contribute to improving the proposed SRWarp method.
- Compared with existing methods, our SRWarp model reconstruct high-quality details and edges in transformed images, quantitatively and qualitatively.

2. Related Work

Conventional deep SR. After Dong *et al.* [8] has successfully applied CNNs to the SR task, numerous approaches have been studied toward better reconstruction. VDSR [20] is one of the most influential works which introduces a novel residual learning strategy to enable faster training and very deep SR network architecture. ESPCN [33] constructs an efficient pixel-shuffling layer to implement a learnable upsampling module. LapSRN [22] architecture efficiently handles the multiscale SR task using a laplacian upscaling pyramid. Ledig *et al.* [25] have adopted the residual block from high-level image classification task [14] to implement SRResNet and SRGAN models. With increasing computational resources, state-of-the-art methods such as EDSR [27] have focused on larger and more complex network structures, producing high quality images. Recently, several advances in neural network designs such as attention [51, 7, 32], back-projection [12], and dense connections [52, 53, 12, 39] have made it possible to reconstruct high-quality images very efficiently.

SR for arbitrary resolution. Most conventional SR methods [8, 20] have relied on naïve interpolation to enlarge a given LR image before Shi *et al.* [33] have introduced the pixel-shuffling layer for learnable upscaling. For example, VDSR [20] upscales the LR image to their target resolution and then applies the SR model to refine local details and textures so that the method can serve as an arbitrary-scale SR framework. However, a significant drawback is that extensive computations are required proportional to the output size. Therefore, subsequent methods have been specialized for some fixed integer scales, e.g., $\times 2$ or $\times 4$, which are commonly used in various applications.

Recently, Hu *et al.* [15] have proposed the meta-upscale module to replace scale-specific upsampling layers in previous approaches. Meta-SR [15] is designed to utilize dynamic filters to deal with real-valued upscaling factors. Subsequently, Wang *et al.* [37] introduce scale-aware features and upsampling modules to reconstruct images of arbitrary target resolutions. The previous methods mainly consider the SR task along horizontal or vertical axes. However, our differentiable warping module in SRWarp allows images to be transformed into any shape.

Irregular spatial sampling in CNNs. Pixels in a digital image are uniformly placed on a 2D rectangle. However, objects may appear in arbitrary shapes and orientations in the image, making it challenging to handle them with a simple convolution. To overcome the limitation, the spatial transformer networks [16] estimate appropriate warping parameters to compensate for possible deformations in the input image. Rather than transform the image, deformable convolutions [6] and active convolutions [17] predict input-dependent kernel offsets and modulators [54] to perform irregular spatial sampling. Furthermore, the deformable kernel [10] approach resamples filter weights to adjust the effective receptive field adaptively. Recent state-of-the-art image restoration models, especially with temporal data, introduce the irregular sampling strategy for accurate alignment [36, 38, 43]. However, our approach is the first novel attempt to interpret image warping as an SR problem.

3. Method

We introduce our generalized SR framework, namely SRWarp, in detail. $\mathbf{I}_{\text{LR}} \in \mathbb{R}^{H \times W}$, $\mathbf{I}_{\text{HR}} \in \mathbb{R}^{H' \times W'}$, and $\mathbf{I}_{\text{SR}} \in \mathbb{R}^{H' \times W'}$ represent source LR, ground-truth HR, and target super-resolved images, respectively. $H \times W$ and $H' \times W'$ correspond to image resolutions, and we omit

RGB color channels for simplicity. Different from the conventional SR, the target resolution $H' \times W'$ varies depending on the given transformation. We define its resolution using a bounding box of the image because the warping may produce irregular output shapes rather than rectangular. For more detailed descriptions and analysis of this section, please refer to our supplementary material.

3.1. Super-Resolution Under Homography

Given a 3×3 projective homography matrix M and a point $p = (x, y, 1)^\mathsf{T}$ on the source image, we calculate the target homogeneous coordinate $p' = w'(x', y', 1)^\mathsf{T}$ as $Mp = p'$ or $f_M(x, y) = (x', y')$, where f_M is a corresponding functional representation. In the backward warping, $p = M^{-1}p'$ is calculated instead for each output pixel p' to remove cavities. If we simply scale the image along x and y axes, then the matrix M is defined as follows:

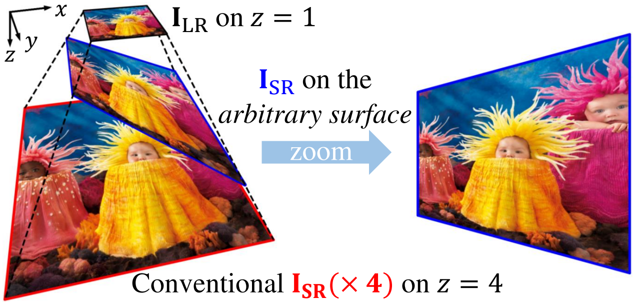
$$M_{s_x s_y} = \begin{pmatrix} s_x & 0 & 0.5\,(s_x - 1) \\ 0 & s_y & 0.5\,(s_y - 1) \\ 0 & 0 & 1 \end{pmatrix}, \tag{1}$$

where s_x and s_y correspond to scale factors along the axes, and the translation components, i.e., $0.5\,(s_* - 1)$, compensate subpixel shift to ensure an accurate alignment [37]. Most early SR methods are designed to deal with the case where $s_x = s_y$ represents predefined integers [33, 25, 12, 39]. Recent approaches have relaxed such constraint by allowing arbitrary real numbers [15, 37]. However, numerous possible forms of M remain unexplored.

Figure 2 presents a concept of the conventional SR methods and our SRWarp method from the perspective of the image scale pyramid. For convenience, we assume that an LR image $\mathbf{I}_{\text{LR}}$ is placed on the plane $z = 1$. Then, obtaining the $\times s$ SR result $\mathbf{I}_{\text{SR}}$ is equivalent to slicing the pyramid by the plane $z = s$, where all points on the SR image have the same z-coordinate. Previous methods aim to learn the image representations that are parallel to the given LR input. However, slicing the image pyramid with an arbitrary plane, or even general surfaces in the space, is also possible. Therefore, We propose to redefine the warping problem as a generalized SR task with spatially-varying scales and even aspect ratios because the pixels in the resulting image can have different z values depending on their positions.

3.2. Adaptive Warping Layer

Image warping consists of two primitive operations, namely, mapping and resampling. The mapping initially determines the spatial relationship between input and output images. For a target position $p' = (x', y')$, the corresponding source pixel is located as $p = (x, y) = f_M^{-1}(x', y')$. We omit the homogeneous representation for simplicity. While pixels in digital images are placed on integer coordinates only, x and y may have arbitrary real values depending on the function f_M^{-1}. Therefore, an appropriate resampling is required to obtain a plausible pixel value as follows:



x
y
z
$\mathbf{I}_{\text{LR}}$ on $z = 1$
$\mathbf{I}_{\text{SR}}$ on the *arbitrary surface*
zoom
Conventional $\mathbf{I}_{\text{SR}}(\times 4)$ on $z = 4$

Figure 2: **Concept of the generalized SR.** Dotted lines represent the upscale image pyramid in xyz-coordinate. While $\times s$ super-resolved images from conventional methods (**red**) exist on the plane $z = s$ only, our results (**blue**) exists on any arbitrary cutting surfaces of the pyramid. We note that the aspect ratio of the pyramid can be varied as well.

$$\mathbf{W}(x', y') = \sum_{i,j=a}^{b} \mathbf{k}(x', y', i, j)\, \mathbf{F}(\lfloor x \rceil + i, \lfloor y \rceil + j), \tag{2}$$

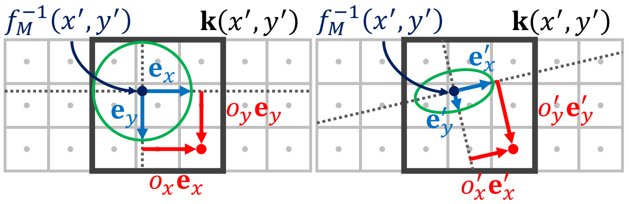
where $\lfloor \cdot \rceil$ is a rounding operator, $\mathbf{F} \in \mathbb{R}^{H \times W}$ is an input, $\mathbf{W} \in \mathbb{R}^{H' \times W'}$ is an output, and $\mathbf{k}$ is a point-wise interpolation kernel, respectively. a and b are boundary indices of the $k \times k$ window, where $k = b - a + 1$. For example, we set $a = -1$ and $b = 1$ for standard 3×3 kernels.

Conventional resampling algorithms introduce a fixed sampling coordinate and kernel function to calculate the weight $\mathbf{k}$, regardless of the transformation M. For example, a widely-used bicubic warping initially calculates a relative offset (o_x, o_y) of each point in the $k \times k$ window with respect to (x, y) as shown in Figure 3a and constructs $\mathbf{k}$ using a cubic spline. However, due to the diversity of possible transformations, such formulation may not be optimal in several aspects. First, it is difficult to consider the transformed geometry where the target image is not defined on a rectangular grid. Second, the fixed kernel function limits its generalizability, while recent SR models prefer learnable upsampling [33, 15, 37] rather than the predetermined one [20]. To handle these issues, we propose an adaptive warping layer (AWL) so that the resampling kernel $\mathbf{k}$ can be trained to consider local deformations.

To determine an appropriate sampling coordinate for each target position (x', y'), we linearize the backward mapping M^{-1} at the point with the Jacobian $J(x', y') = \begin{pmatrix} \mathbf{u}^\mathsf{T} & \mathbf{v}^\mathsf{T} \end{pmatrix}$. Specifically, we calculate $\mathbf{u}$ and $\mathbf{v}$ as follows:

$$\begin{aligned} \mathbf{u} &= \frac{f_M^{-1}(x' + \epsilon, y') - f_M^{-1}(x' - \epsilon, y')}{2\epsilon}, \\ \mathbf{v} &= \frac{f_M^{-1}(x', y' + \epsilon) - f_M^{-1}(x', y' - \epsilon)}{2\epsilon}, \end{aligned} \tag{3}$$

where $f_M^{-1} = f_{M^{-1}}$ and $\epsilon = 0.5$. We project a unit cir-



(a) Regular resampling (b) Adaptive resampling

Figure 3: **Example of the adaptive grid.** Each point represents a pixel on the source domain $\mathbf{F}$. (a) On a regular grid, the resampling bases $\mathbf{e}_x$ and $\mathbf{e}_y$ in **blue** are orthonormal and aligned with the source image. (b) We adopt irregular bases $\mathbf{e}'_x$ and $\mathbf{e}'_y$ with varying lengths and orientations for each target position (x', y'). The relative offset vector (o_x, o_y) of an example point in **red** is mapped to $\left(o'_x, o'_y\right)$ following the change of basis. The **green** ellipse illustrates how a unit circle on the target image is projected to the source domain.

cle centered around (x', y') on the target domain to an ellipse [11] on the source image, using the local approximation. Then, we calculate two principal axes $\mathbf{e}'_x$ and $\mathbf{e}'_y$ of the ellipse. As shown in Figure 3b, the relative offset vector $\mathbf{o} = (o_x, o_y)$ is represented as $\mathbf{o}' = \left(o'_x, o'_y\right)$ under the new locally adaptive coordinate system. In the resampling process, the actual contribution of each point is calculated with respect to the distance from the origin. Therefore, we utilize the adaptive coordinate to adjust the point considering local distortions. The original offset vectors are rescaled as $\frac{\|\mathbf{o}'\|}{\|\mathbf{o}\|}\mathbf{o}$ and used to calculate the kernel $\mathbf{k}$.

Subsequently, we introduce a kernel estimator $\mathcal{K}$ to estimate adaptive resampling weights $\mathbf{k}$. Similar to the conventional interpolation functions, it takes k^2 offset vectors to determine the contributions of each point $\mathbf{F}(x, y)$ in the window $\mathbf{k}(x', y')$. However, we adopt a series of fully-connected layers [15, 37] to learn the function rather than using a predetermined one. The learnable network allows considering local deformations and generates appropriate dynamic filters [18, 19] for a given transformation. We construct the proposed AWL $\mathcal{W}$ by combining the adaptive resampling grid and kernel prediction layer $\mathcal{K}$, as follows:

$$\mathbf{W} = \mathcal{W}(\mathbf{F}, f_M). \tag{4}$$

3.3. Multiscale Blending

Figure 2 illustrates that images under the generalized SR task suffer distortions with spatially-varying scaling factors. Therefore, multiscale representations can play an essential role in reconstructing high-quality images. To effectively utilize the property, we further introduce a blending method for the proposed SRWarp framework.

Multiscale feature extractor. We define a scale-specific feature extractor $\mathcal{F}_{\times s}$ with a fixed integer scaling factor of s, which adopts the state-of-the-art SR architectures [27, 39]. Given an LR image $\mathbf{I}_{\text{LR}}$, the module extracts the scale-specific feature $\mathbf{F}_{\times s} \in \mathbb{R}^{C \times sH \times sW}$, where C denotes the number of output channels. While it is possible to separate the network for each scaling factor, we adopt a shared feature extractor with multiple upsampling layers [27] in practice for several reasons. For instance, previous approaches have demonstrated that multiscale representations can be jointly learned [20, 22, 23, 37] within a single model. Also, using the shared backbone network is computationally efficient compared to applying multiple different models to extract spatial features. From the state-of-the-art SR architecture, we replace the last upsampling module with ×1, ×2, and ×4 feature extractor to implement our multiscale backbone as shown in Figure 4.

Multiscale warping and blending. For each scale-specific feature $\mathbf{F}_{\times s}$, we construct the corresponding transformation as $MM_{s^{-1}s^{-1}}$ by using (1). As a result, features of different resolutions can be mapped to a fixed spatial dimension, i.e., $H' \times W'$. We use a term $\mathbf{W}_{\times s} \in \mathbb{R}^{C \times H' \times W'}$ to represent the warped features as follows:

$$\mathbf{W}_{\times s} = \mathcal{W}\left(\mathcal{F}_{\times s}\left(\mathbf{I}_{\text{LR}}\right), MM_{s^{-1}s^{-1}}\right). \tag{5}$$

Then, the output SR image $\mathbf{I}_{\text{SR}}$ can be reconstructed from a set of the multiscale warped features $\{\mathbf{W}_{\times s} | s = s_0, s_1, \cdots\}$. However, a simple combination, e.g., averaging or concatenation, of those features may not reflect the spatially-varying property of the generalized SR problem. Therefore, we introduce a multiscale blending module to combine information from different resolutions effectively. To determine appropriate scales for each local region, image contents play a critical role. For example, low-frequency components are preferred in the warping process to prevent aliasing and undesirable artifacts for plain regions. On the contrary, high-frequency details are considered to represent edges and textures accurately. Therefore, we use learnable scale-specific and global content feature extractors $\mathcal{C}_{\times s}$ and $\mathcal{C}$ as follows:

$$\mathbf{C} = \mathcal{C}\left(\mathcal{C}_{\times s_0}\left(\mathbf{W}_{\times s_0}\right), \mathcal{C}_{\times s_1}\left(\mathbf{W}_{\times s_1}\right), \cdots\right), \tag{6}$$

where the global content feature $\mathbf{C} \in \mathbb{R}^{C \times H' \times W'}$ is represented by scale-specific representations $\mathcal{C}_{\times s_i}\left(\mathbf{W}_{\times s_i}\right)$.

Since our SRWarp method handles spatially-varying distortions, appropriate feature scales may also depend on the local deformation. The proposed model may benefit from the degree of transformation around the pixel to determine the contributions of each multiscale representation. Therefore, we acquire the scale feature $\mathbf{S} \in \mathbb{R}^{H' \times W'}$ as follows:

$$\mathbf{S}(x', y') = -\log\left|\det\left(J(x', y')\right)\right|. \tag{7}$$

Physically, the determinant of the Jacobian describes a local magnification factor of the transformation. When adopting backward mapping, we consider the reciprocal of the Jacobian determinant and normalize it by taking the natural log.

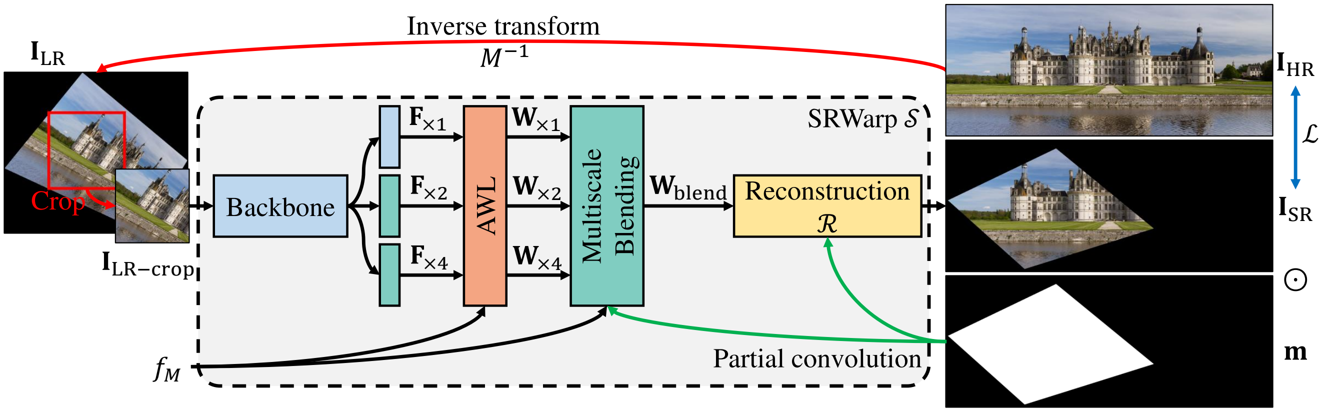


Figure 4: **Overall organization of the proposed SRWarp model.** More detailed architectures are described in our supplementary material. Black regions outside the warped image $\mathbf{I}_{\text{SR}}$ represent void pixels that are ignored.

Our multiscale blending module then applies 1×1 convolutions to the concatenated content and scale features $\mathbf{C}$ and $\mathbf{S}$. By doing so, appropriate blending weights $w_{\times s}$ are determined for each output position (x', y'). The blended features $\mathbf{W}_{\text{blend}}$ can be represented as follows:

$$\mathbf{W}_{\text{blend}} = \sum_s w_{\times s} \odot \mathbf{W}_{\times s}, \tag{8}$$

where $\odot$ is an element-wise multiplication.

Partial convolution. Image warping can produce void pixels on the target coordinate when the point is mapped to outside the source image. Such regions may negatively affect the model performance because conventional CNNs consider all pixels equally. To efficiently deal with the problem, we define a 2D binary mask $\mathbf{m}$ as follows:

$$\mathbf{m}(x', y') = \begin{cases} 0, & \text{if } (x, y) \text{ is outside of } \mathbf{F}_{\times 1}, \\ 1, & \text{otherwise,} \end{cases} \tag{9}$$

where $f_M(x, y) = (x', y')$. We calculate the mask $\mathbf{m}$ from the $\times 1$ feature $\mathbf{F}_{\times 1}$ and share it across scales to maintain consistency between different resolutions. Then, we adopt the partial convolution [28, 29] for our content feature extractors $\mathcal{C}$ and $\mathcal{C}_{\times s}$, using the mask to ignore void pixels.

3.4. SRWarp

Finally, we introduce a reconstruction module $\mathcal{R}$ with five residual blocks [25, 27]. We combine the SR backbone, AWL, blending, and reconstruction modules to construct the SRWarp model $\mathcal{S}$ as shown in Figure 4. For stable training, the residual connection [20] is incorporated as $\mathbf{I}_{\text{SR}} = \mathcal{R}(\mathbf{W}_{\text{blend}}) + \mathbf{I}_{\text{bic}}$, where $\mathbf{I}_{\text{bic}}$ is a warped image using bicubic interpolation and $\mathbf{I}_{\text{SR}}$ is the final output. Given a set of training input and target pairs $(\mathbf{I}^n_{\text{LR}}, \mathbf{I}^n_{\text{HR}})$, we minimize an average L_1 loss [22, 27] $\mathcal{L}$ between the reconstructed and ground-truth (GT) images as follows:

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{\|\mathbf{m}\|_0} \left\| \mathbf{m} \odot \left(\mathcal{S}(\mathbf{I}^n_{\text{LR}}, f_M) - \mathbf{I}^n_{\text{HR}} \right) \right\|_1, \tag{10}$$

where $N = 4$ is the number of samples in a mini-batch, n is a sample index, 0-norm $\|\cdot\|_0$ represents the number of nonzero values, and $\mathcal{S}(\mathbf{I}^n_{\text{LR}}, f_M) + \mathbf{I}^n_{\text{bic}} = \mathbf{I}^n_{\text{SR}}$, respectively. The transform function f_M is shared in a single mini-batch for efficient calculation. The binary mask $\mathbf{m}$ in (9) prevents backward gradients from being propagated from void pixels. The proposed SRWarp model can be trained in an end-to-end manner with the ADAM [21] optimizer.

4. Experiments

We adopt two different SR networks as a backbone of the multiscale feature extractor for the proposed SRWarp model. The modified MDSR [27] architecture serves as a smaller baseline, whereas RRDB [39] with customized multiscale branches (MRDB) provides a larger backbone for improved performance. We describe more detailed training arguments in our supplementary material. PyTorch codes with an efficient CUDA implementation and dataset will be publicly available from the following repository: https://github.com/sanghyun-son/srwarp.

4.1. Dataset and Metric

Dataset. In conventional image SR methods, acquiring real-world LR and HR image pairs is very challenging due to several practical issues, such as outdoor scene dynamics and subpixel misalignments [4, 5, 50]. Similarly, collecting high-quality image pairs with corresponding transformation matrices in the wild for our generalized SR task is also difficult. Therefore, we propose the DIV2K-Warping (DIV2KW) dataset by synthesizing LR samples from the existing DIV2K [1] dataset to train our SRWarp model in a supervised manner. We first assign 500, 100, and 100 random warping parameters $\{M_i\}$ for training, validation, and test, respectively. Each matrix is designed to include random upscaling, sheering, rotation, and projection because we mainly aim to enlarge the given image. We describe more details in our supplementary material.

During the learning phase, we randomly sample square HR patches from 800 images in the DIV2K training dataset and one warping matrix M_i^{-1} to construct a ground-truth batch $\mathbf{I}_{\text{HR}}$. Then, we warp the batch with M_i^{-1} to obtain corresponding LR inputs $\mathbf{I}_{\text{LR}}$. For efficiency, the largest valid square from the transformed region is cropped for the input $\mathbf{I}_{\text{LR-crop}}$. With the transformation matrix M_i and LR patches $\mathbf{I}_{\text{LR-crop}}$, we optimize our warping model to reconstruct the original image $\mathbf{I}_{\text{HR}}$ as described in 3.4. Figure 4 illustrates the actual training pipeline regarding our SRWarp model. We use 100 images from the DIV2K valid dataset with different transformation parameters following the same pipeline to evaluate our method.

Metric. We adopt a traditional PSNR metric on RGB color space to evaluate the quality of warped images. However, we only consider valid pixels in a $H' \times W'$ grid similar to our training objective in (10) because they have irregular shapes rather than standard rectangles. The modified PSNR with a binary mask $\mathbf{m}$ (mPSNR) is described as follows:

$$\text{mPSNR(dB)} = 10 \log_{10} \frac{\|\mathbf{m}\|_0}{\|\mathbf{m} \odot (\mathbf{I}_{\text{SR}} - \mathbf{I}_{\text{HR}})\|_2^2}, \qquad (11)$$

where images $\mathbf{I}_*$ are normalized between 0 and 1.

4.2. Ablation Study

We extensively validate possible combinations of the proposed modules in Table 1 because they are orthogonal to each other. The modified baseline MDSR [27] structure is used as a backbone by default for lightweight evaluations. We refer to the model with a single-scale SR backbone [27, 39] and standard warping layer at the end as a baseline. Sequences of A, M, and R represent corresponding configurations, e.g., A-R for the one which shows 32.19dB mPSNR in Table 1. Our SRWarp model is represented as A-M-R and achieves 32.29dB in Table 1. More details are described in our supplementary material.

Adaptive warping layer. Table 1 demonstrates that AWL introduces consistent performance gains by providing spatially-adaptive resampling kernels. Table 2a extensively compares possible implementations of the AWL in the proposed SRWarp method. We replace the regular resampling grid from M-R in Table 1 to the spatially-varying representations (Adaptive in Table 2a) as shown in Figure 3b. However, because the resampling weights $\mathbf{k}$ are not learnable, the formulation does not bring an advantage even when the spatially-varying property is considered. Introducing a kernel estimator to the regular grid (Layer in Table 2b) yields +0.06dB of mPSNR gain over the M-R method. The performance is further improved to 32.29dB (A-M-R in Table 1) by combining the spatially-varying coordinates and trainable module. We note that the adaptive resampling grid at each output position does not require any additional parameters and can be calculated efficiently.

A	M	R	B	mPSNR↑(dB) on DIV2KW$_{\text{Valid.}}$
–	–	–	EDSR	31.36 (+0.00)
✓	–	–		32.06 (+0.70)
–	–	✓		32.08 (+0.72)
✓	–	✓		32.19 (+0.83)
–	✓	–	MDSR	32.19 (+0.83)
✓	✓	–		32.21 (+0.85)
–	✓	✓		32.19 (+0.83)
✓	✓	✓		**32.29 (+0.93)**
–	–	–	RRDB	31.64 (+0.28)
✓	✓	✓	MRDB	**32.56 (+1.20)**

Table 1: **Contributions of each module in our SRWarp method.** A, M, R, and B denote the adaptive warping layer (AWL), multiscale blending, reconstruction module, and backbone architecture, respectively. Numbers in parentheses indicate performance gains over the baseline on top.

Method	mPSNR↑(dB)
Adaptive	32.19
Layer	32.25
AWL-SS	32.23
AWL-MS	32.24

(a) Warping

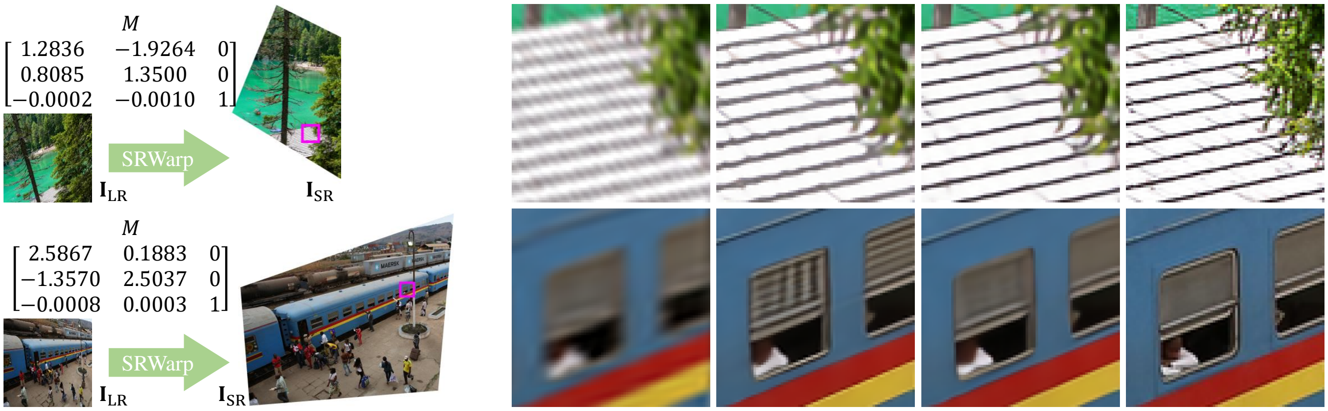
Method	mPSNR↑(dB)
Average	32.26
Concat.	32.19
w/o **C**	32.24
w/o **S**	32.23

(b) Blending

Table 2: **Effects of warping and blending strategies in our SRWarp model.** We evaluate each method on the DIV2KW$_{\text{Valid.}}$ dataset. The proposed SRWarp achieves the mPSNR of 32.29dB under the same environment.

We also analyze two possible variants of our AWL. AWL-SS in Table 2a shares the kernel estimator $\mathcal{K}$ across scales and channel dimensions, even with the multiscale SR backbone. AWL-MS in Table 2a achives a minor performance gain of +0.01dB by utilizing scale-specific modules $\mathcal{K}_{\times s}$ as described in Section 3.3. In the proposed SRWarp model, we further estimate the kernels in a depthwise manner, i.e., $C \times k \times k$ weights for each (x', y'), and achieve an additional +0.05dB improvement in the mPSNR metric.

Multiscale blending. Table 1 shows that our multiscale approach (M) consistently improves the spatially-varying SR performance by a larger margin than the single-scale counterpart with the baseline EDSR [27] backbone (A-R in Table 1). We justify the design of our blending module in Table 2b by only changing the formulation to calculate the combination coefficients $w_{\times s}$ in (8). Interestingly, simply averaging (Average in Table 2b) the warped features $\mathbf{W}_{\times s}$ produces a better result than concatenating and blending them with a trainable 1×1 convolutional layer (Concat. in Table 2b). Such performance decrease demonstrates that an appropriate design is required for the efficient blending module because concatenation is a more general formulation. We also analyze how content and scale features sup-



(a) $\mathbf{I}_{\text{LR}}$, $\mathbf{I}_{\text{SR}}$, and M (b) `cv2` (c) RRDB + `cv2` (d) **SRWarp** (e) GT

Figure 5: **Qualitative warping results on the DIV2KW$_{\text{Test}}$ dataset.** We provide input LR and output HR images with corresponding warping matrices M_i. Translation components are omitted for simplicity. Patches are cropped from the DIV2KW$_{\text{Test}}$ '*0807.png*' and '*0850.png*.' More visual comparisons are included in our supplementary material.

Method			mPSNR$^{\uparrow}$(dB) on DIV2KW$_{\text{Test}}$
`cv2` (Bicubic) [3]			27.85 (-2.41)
×2	RDN [52]	+`cv2`	30.22 (+0.00)
	EDSR [27]		30.42 (+0.20)
	RCAN [51]		30.45 (+0.23)
×4	RDN [52]	+`cv2`	30.50 (+0.28)
	EDSR [27]		30.66 (+0.44)
	RCAN [51]		30.71 (+0.49)
	RRDB [39]		<u>30.76 (+0.54)</u>
SRWarp (MRDB)			**31.04 (+0.82)**

Table 3: **Comparison between our SRWarp and available warping methods.** + `cv2` denotes that we first apply a scale-specific SR model for supersampling and then transform the upscaled image with the traditional warping algorithm. Numbers in parenthesis denote performance gain over the ×2 RDN + `cv2` method. The best and second-best performances are **bolded** and <u>underlined</u>, respectively.

port the blending module to combine multiscale representations effectively. If the content information is ignored (w/o **C** in Table 2b), our SRWarp model suffers an mPSNR drop of 0.05dB. Removing the scale features (w/o **S** in Table 2b) also brings a similar degree of performance degradation, justifying the design of the proposed multiscale blending.

Reconstruction module and partial convolution. Since the reconstruction unit $\mathcal{R}$ can further refine output images, the module evidently brings additional performance gains for all combinations ($*$-R in Table 1). We also examine the usefulness of the partial convolution [28, 29] in the content feature extractor and reconstruction module. Compared to the previous SR methods, our SRWarp framework is more sensitive to boundary effects due to several reasons. First, image boundaries, i.e., regions between valid and void areas, are not aligned with convolutional kernels and have irregular shapes. Second, because we place irregular-shaped data on a regular 2D grid, numerous void pixels in the warped image negatively affect the following learnable layers. SRWarp converges much slower without the partial convolution, and its final performance decreases by 0.06dB due to the severe boundary effects.

Backbone architecture. The last two rows of Table 1 show the effects of different backbone architectures on the performance of the SRWarp method. Using the larger MRDB network with 17.1M parameters results in a significant PSNR gain of +0.27dB compared with the MDSR backbone with 1.7M parameters, indicating better fitting on the training data results in higher validation performance.

4.3. Comparison with the Other Methods

We compare the proposed SRWarp with existing methods. We note that providing an exact comparison with other methods is difficult given that our approach is the *first* attempt toward generalized image SR. First, we adopt a conventional interpolation-based warping algorithm from the OpenCV [3]. We use `cv2.WarpPerspective` function with a bicubic kernel to synthesize warped images. For alternatives, we combine state-of-the-art SR models and the traditional warping operation. Since the given LR images are supersampled before interpolation, the warping function can synthesize high-quality results directly. We note that the transformation matrix M is compensated to $MM_{s^{-1}s^{-1}}$ for $\times s$ SR model because the outputs from SR models are $\times s$ larger than the original input.

Table 3 provides quantitative comparison of various methods. For fairness, we adopt the DIV2KW$_{\text{Test}}$ dataset rather than the validation split used in Section 4.2. Com-

			PSNR$^\uparrow$(dB) on B100 [31] with arbitrary scale factors								
Method	# Params	Runtime	×2.0	×2.2	×2.5	×2.8	×3.0	×3.2	×3.5	×3.8	×4.0
SRCNN [8]	0.06M	2340ms	27.11	27.85	28.62	28.71	28.37	27.89	27.17	26.59	26.27
VDSR [20]	0.67M	26ms	31.82	30.36	29.54	28.84	28.77	28.15	27.82	27.46	27.27
Meta-EDSR [15]	40.1M	218ms	32.26	31.31	30.40	29.61	29.22	28.82	28.27	27.86	27.67
Meta-RDN [15]	22.4M	253ms	**32.33**	31.45	**30.46**	29.69	29.26	28.88	28.41	**28.01**	27.71
SRWarp (MRDB)	18.3M	155ms	32.31	**31.46**	**30.46**	**29.71**	**29.27**	**28.89**	**28.42**	**28.01**	**27.77**

Table 4: **Quantitative comparison of the arbitrary-scale SR task.** We use official implementations of each method and compare them in a unified environment. The average runtime is measured on the ×3.0 SR task using 100 test images excluding initialization, I/O, and the other overheads. For the SRCNN [8] model, the ×3 network (9-5-5) is evaluated across all scaling factors [20] on CPU. Our SRWarp consistently outperforms the other approaches even with fewer parameters.

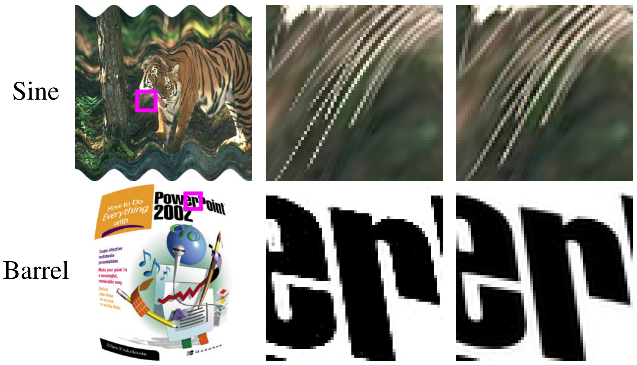
pared with the traditional `cv2` algorithm, using the SR methods provides significant improvements with the mPSNR gain of at least +2.41dB. Higher-scale ×4 models tend to perform better than their ×2 counterparts, justifying the importance of the fine-grained supersampling. Our SRWarp model further outperforms the other SR-based formulations by a large margin. Figure 5 shows that our approach reconstructs much sharper images with less aliasing, demonstrating the effectiveness of AWL and multiscale blending.

4.4. Arbitrary-scale SR

Our SRWarp provides a generalization of the conventional SR models defined on scaling matrices in (1) only. To justify that the proposed framework is compatible with existing formulations, we evaluate our method on the regular SR tasks of arbitrary scaling factors. We train our SRWarp model with RRDB [39] backbone on fractional-scale DIV2K dataset [15] and evaluate it following Hu *et al.* [15]. The input transformation of the SRWarp is constrained to (1), and the other configurations are fixed. Table 4 shows an average PSNR of the luminance (Y) channel between SR results and ground-truth images on B100 [31] dataset. We note that SRCNN [8] and VDSR [20] first resize the input image to an arbitrary target resolution before take it into the network. Compared with the meta-upscale module (Meta-EDSR and Meta-RDN), our adaptive warping layer and multiscale blending provide an efficient and generalized model for the arbitrary-scale SR task.

4.5. Over the Homographic Transformation

Our SRWarp model is trained on homographic transformation only. However, we can extend the method to an arbitrary backward mapping equation in a functional form $f_M^{-1}(x', y') = (x, y)$ without *any* modification. Although we adopt homographic transformations only for the training, the adaptive warping layer and multiscale blending help the method be generalized well on unseen deformations. Figure 6 compares our results on various functional transforms against the combination of RRDB [39] and traditional bicubic interpolation. Our SRWarp model can provide more flexibility and diversity in general image editing tasks by reconstructing visually pleasing edge structures.



(a) $\mathbf{I}_{\text{SR}}$ (b) RRDB [39] (c) **SRWarp**

Figure 6: **General image warping with our SRWarp.** We apply various functional transforms to samples from B100 [31] '*108005.png*' and Set14 [46] '*ppt3.png*.' (b) RRDB corresponds to RRDB + `cv2` in Table 3.

5. Conclusion

We propose a generalization of the conventional SR tasks under image transformation for the first time. Our SRWarp framework deals with the spatially-varying upsampling task when arbitrary resolutions and shapes are required to the output image. We also provide extensive ablation studies on the proposed method to validate the contributions of several novel components, e.g., adaptive warping layer and multiscale blending, in our design. The visual comparison demonstrates why the SRWarp model is required for image warping, justifying the advantage of the proposed method.

Acknowledgement

This work was supported by IITP grant funded by the Ministry of Science and ICT of Korea (No. 2017-0-01780) and AIRS Company in Hyundai Motor Company & Kia Motors Corporation through HKMC-SNU AI Consortium Fund.

References

[1] Eirikur Agustsson and Radu Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *CVPR Workshops*, 2017.

[2] Yuval Bahat and Tomer Michaeli. Explorable super resolution. In *CVPR*, 2020.

[3] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[4] Jianrui Cai, Hui Zeng, Hongwei Yong, Zisheng Cao, and Lei Zhang. Toward real-world single image super-resolution: A new benchmark and a new model. In *ICCV*, 2019.

[5] Chang Chen, Zhiwei Xiong, Xinmei Tian, Zheng-Jun Zha, and Feng Wu. Camera lens super-resolution. In *CVPR*, 2019.

[6] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017.

[7] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *CVPR*, 2019.

[8] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *TPAMI*, 2016.

[9] Ying Fu, Tao Zhang, Yinqiang Zheng, Debing Zhang, and Hua Huang. Hyperspectral image super-resolution with optimized RGB guidance. In *CVPR*, 2019.

[10] Hang Gao, Xizhou Zhu, Steve Lin, and Jifeng Dai. Deformable kernels: Adapting effective receptive fields for object deformation. In *ICLR*, 2020.

[11] N. Greene and P. S. Heckbert. Creating raster omnimax images from multiple perspective views using the elliptical weighted average filter. *CG & A*, 6(6):21–27, 1986.

[12] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Deep back-projection networks for super-resolution. In *CVPR*, 2018.

[13] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita. Space-time-aware multi-resolution video enhancement. In *CVPR*, 2020.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[15] Xuecai Hu, Haoyuan Mu, Xiangyu Zhang, Zilei Wang, Tieniu Tan, and Jian Sun. Meta-SR: A magnification-arbitrary network for super-resolution. In *CVPR*, 2019.

[16] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *NIPS*, 2015.

[17] Yunho Jeon and Junmo Kim. Active convolution: Learning the shape of convolution for image classification. In *CVPR*, 2017.

[18] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. In *NIPS*, 2016.

[19] Younghyun Jo, Seoung Wug Oh, Jaeyeon Kang, and Seon Joo Kim. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *CVPR*, 2018.

[20] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, 2016.

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[22] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *CVPR*, 2017.

[23] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Fast and accurate image super-resolution with deep laplacian pyramid networks. *TPAMI*, 41(11):2599–2613, 2018.

[24] Hoang Le, Feng Liu, Shu Zhang, and Aseem Agarwala. Deep homography estimation for dynamic scenes. In *CVPR*, 2020.

[25] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017.

[26] Wenbo Li, Xin Tao, Taian Guo, Lu Qi, Jiangbo Lu, and Jiaya Jia. MuCAN: Multi-correspondence aggregation network for video super-resolution. In *ECCV*, 2020.

[27] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPR Workshops*, 2017.

[28] Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *ECCV*, 2018.

[29] Guilin Liu, Kevin J. Shih, Ting-Chun Wang, Fitsum A. Reda, Karan Sapra, Zhiding Yu, Andrew Tao, and Bryan Catanzaro. Partial convolution based padding. In *arXiv*, 2018.

[30] Andreas Lugmayr, Martin Danelljan, Luc Van Gool, and Radu Timofte. SRFlow: Learning the super-resolution space with normalizing flow. In *ECCV*, 2020.

[31] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, 2001.

[32] Ben Niu, Weilei Wen, Wenqi Ren, Xiangde Zhang, Lianping Yang, Shuzhen Wang, Kaihao Zhang, Xiaochun Cao, and Haifeng Shen. Single image super-resolution via a holistic attention network. In *ECCV*, 2020.

[33] Wenzhe Shi, Jose Caballero, Ferenc Huszar, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR*, 2016.

[34] Rahul Swaminathan and Shree K Nayar. Nonmetric calibration of wide-angle lenses and polycameras. *TPAMI*, 22(10):1172–1178, 2000.

[35] Yang Tan, Haitian Zheng, Yinheng Zhu, Xiaoyun Yuan, Xing Lin, David Brady, and Lu Fang. CrossNet++: Cross-scale large-parallax warping for reference-based super-resolution. *TPAMI*, 2020.

[36] Yapeng Tian, Yulun Zhang, Yun Fu, and Chenliang Xu. TDAN: Temporally-deformable alignment network for video super-resolution. In *CVPR*, 2020.

[37] Longguang Wang, Yingqian Wang, Zaiping Lin Lin, Jungang Yang, Wei An, and Yulan Guo. Learning for scale-arbitrary super-resolution from scale-specific networks. *arXiv*, 2020.

[38] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. EDVR: Video restoration with enhanced deformable convolutional networks. In *CVPRW*, 2019.

[39] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. ESRGAN: enhanced super-resolution generative adversarial networks. In *ECCV Workshops*, 2018.

[40] Yingqian Wang, Longguang Wang, Jungang Yang, Wei An, Jingyi Yu, and Yulan Guo. Spatial-angular interaction for light field image super-resolution. In *ECCV*, 2020.

[41] Pengxu Wei, Ziwei Xie, Hannan Lu, ZongYuan Zhan, Qixiang Ye, Wangmeng Zuo, and Liang Lin. Component divide-and-conquer for real-world image super-resolution. In *ECCV*, 2020.

[42] Bartlomiej Wronski, Ignacio Garcia-Dorado, Manfred Ernst, Damien Kelly, Michael Krainin, Chia-Kai Liang, Marc Levoy, and Peyman Milanfar. Handheld multi-frame super-resolution. *ACM TOG*, 38(4):1–18, 2019.

[43] Xiaoyu Xiang, Yapeng Tian, Yulun Zhang, Yun Fu, Jan P. Allebach, and Chenliang Xu. Zooming Slow-Mo: Fast and accurate one-stage space-time video super-resolution. In *CVPR*, 2020.

[44] Zhucun Xue, Nan Xue, Gui-Song Xia, and Weiming Shen. Learning to calibrate straight lines for fisheye image rectification. In *CVPR*, 2019.

[45] Jing Yao, Danfeng Hong, Jocelyn Chanussot, Deyu Meng, Xiaoxiang Zhu, and Zongben Xu. Cross-attention in coupled unmixing nets for unsupervised hyperspectral super-resolution. In *ECCV*, 2020.

[46] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces*, 2010.

[47] Jirong Zhang, Chuan Wang, Shuaicheng Liu, Lanpeng Jia, Nianjin Ye, Jue Wang, Ji Zhou, and Jian Sun. Content-aware unsupervised deep homography estimation. In *ECCV*, 2020.

[48] Lei Zhang, Jiangtao Nie, Wei Wei, Yanning Zhang, Shengcai Liao, and Ling Shao. Unsupervised adaptation learning for hyperspectral imagery super-resolution. In *CVPR*, 2020.

[49] Shuo Zhang, Youfang Lin, and Hao Sheng. Residual networks for light field image super-resolution. In *CVPR*, 2019.

[50] Xuaner Zhang, Qifeng Chen, Ren Ng, and Vladlen Koltun. Zoom to learn, learn to zoom. In *CVPR*, 2019.

[51] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, 2018.

[52] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *CVPR*, 2018.

[53] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image restoration. *TPAMI*, 2020.

[54] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable ConvNets v2: More deformable, better results. In *CVPR*, 2019.

SRWarp：基于任意变换的广义图像超分辨率

Sanghyun Son, Kyoung Mu Lee

2021 年 10 月 13 日

摘要

深度卷积神经网络在图像处理及其应用 (包括单帧图像超分辨率) 中效果显著。然而，传统方法仍然受限于预定义的整数比例因子，如：放大 2 倍或 4 倍。因此，难以将传统方法应用于目标分辨率任意的情况。最近的研究通过不同的长宽比处理分辨率的局限性，将比例因子从整数扩大至实数范围。在本论文中，我们提出 SRWarp 框架以扩展超分辨率模型，进而处理图像任意变换的问题。我们将传统的图像扭转任务 (特别当输入图像被放大时) 解释为空间变化的超分辨率问题。同时，我们提出了一些新的公式 (包括自适应扭曲层和多尺度混合)，在转换重建过程中得到视觉满意的效果。和以往的方法相比，我们没有将超分辨率模型限制在规则网格中，相反，我们允许在图像编辑中出现灵活多样的变化。

1 介绍

作为一个基础的视觉问题，图像超分辨率 (SR) 旨在将给定的低分辨率 (LR) 输入图像重建为高分辨率 (HR) 图像。超分辨率方法实用性强，被广泛应用于感知图像增强 [25, 39]、图像编辑 [2, 30] 和数码变焦 [42] 等众多应用中。和其他与视觉相关的任务类似，最近的卷积神经网络在大规模数据集 [1, 27]、规则有效结构 [51, 52, 12] 和新优化技术 [27, 39] 的支持下，达到了优良的超分辨率性能。先进的方法能够在放大 4 倍或 8 倍时，对多种输入数据 (包括现实世界的图像 [50, 4, 41]、视频 [38, 43, 13, 36, 26]、高光谱 [9, 48, 45] 和光场阵列 [49, 35, 40]) 进行超分辨率重建，得到锐利的边缘、清晰的纹理和丰富的细节。

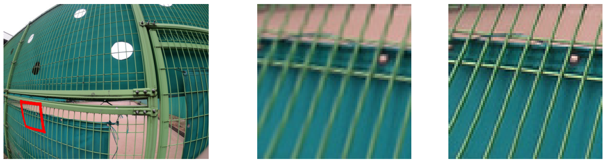
从图像编辑应用程序的角度看，当高分辨率图像 (HR) 难以获取时，超分辨率算法能够有效帮助用户增加图像的像素数量，并重建高频细节。这种操作可能包括使用预定义缩放因子进行缩放或合成任意超分辨率的图像。然而，现有的超分辨率模型只能处理某些整数缩放因子的问题 [25, 51, 39]，因此难以直接将其应用于这些场景中。最近，一些模型支持任意指定缩放因子 [15] 和长宽比 [37]，对给定图像进行上采样操作，以扩大超分辨率模型的应用范围。这些新方法使得现有的超分辨率模型在实际应用中更加灵活和通用。

然而，现有的超分辨率模型受模型自身所限，不能对一般的图像编辑任务进行完全优化。由于数据图像被定义在矩形网格中，先前的方法只能解决这类矩形的图像。然而，在实际的图像编辑中，图像可能经历多种变形，以有效调整图像信息。比如，单应变换 [47, 24] 将不同视角的图像对齐，相机通过多种校正算法消除镜头畸变 [34, 44]。传统扭曲方法的缺点之一是：当图像的局部区域被拉伸时，基于插值的算法会生成较为模糊的结果。因此，超分辨率模型需要有合适的增强算法，能够在对图像进行上采样操作时保留锐利的边缘和详细的纹理。然而，这类应用程序需要在不规则网格上处理图像，因此无法使用处理规则网格图像的传统卷积神经网络。

为避免出现模糊的扭曲结果，在图像变换之前应引入先进的超分辨模型。通过这种方法，超采样像素能够减轻插值法引起的模糊和伪影。然而，胡等人 [15] 和王等人 [37] 已经验证这种方法在任意缩放倍数的超分辨率模型中是次优的。此外，广义扭曲算法应该解决更复杂、甚至是空间变换的变形问题，这在先前的研究中没有直接考虑过。因此，需要一个合适的方法能够将超分辨率模型和扭曲方法有效结合。

在本文的研究中，我们将广义图像变换任务视为空间变换的超分辨率问题。因此，我们建立了端到端的可学习框架 SRWarp。和先前的超分辨率模型不同，我们的方法在两个特定方面解决扭曲问题。首先，我们引入了自适应扭曲层 (AWL)，为不同的局部畸变动态预测合适的重采样核。其次，我们的多尺度混合策略包括基于内容和局部变形的多分辨率特征，以从给定图像中获取更丰富的信息。因此，如图 1 所示，使用有效的主干网路 [27, 39]，我们所提出的 SRWarp 能够对传统扭曲方法忽略的图像结构成功重建。我们的贡献可以总结为以下三点：

- 新模型 SRWarp 对超分辨率的概念进行拓展，使其能够应用于任意变换中，同时提出了能够学习图像变换的框架。
- 分析显示，我们提出的自适应扭曲层和多尺度混合策略有利于提高 SRWarp 模型的性能。
- 和现有模型进行定性和定量的对比，SRWarp 模型对经过变换的图像重建，能得到高质量细节和边缘。



(a) Input (b) OpenCV (c) SRWarp

图 1: **使用 SRWarp 对现实世界的图像进行镜头畸变校正**。图片 1(a)由 GoPro HERO6 手持相机拍摄。SRWarp 支持局部缩放因子的超分辨率重建，可以将输入图像转换为所需形状

2 相关工作

传统的深度超分辨率模型 在董等人 [8] 成功将卷积神经网络应用至超分辨率任务后，已有大量方法能够实现更好的重建效果。VDSR[20] 是其中最有影响力的模型之一，它引入了一种新的残差学习策略，提高了训练速度，并且支持极深的超分辨率网络结构。ESPCN[33] 构建了一个有效的像素变换层，以实现可学习的上采样模块。LapSRN[22] 结构能够通过使用拉普拉斯上采样金字塔有效解决多尺度超分辨率模型。Ledig 等人 [25] 采用高层次图像分类任务 [14] 中的残差模块以实现 SRResNet 和 SRGAN 模型。随着计算资源的增加，先进的方法 (如 EDSR)[27] 通过更大、更复杂的网络结构生成高质量图像。近年来，神经网络中的注意力机制 [51, 7, 32]、直方图反向投影 [12] 和密集连接 [52, 53, 12, 39] 能够有效重建出高质量图像。

适用于任意分辨率的超分辨率模型 大多数传统的超分辨率方法 [8, 20] 依赖于简单的差值方法对低分辨率图像进行方法。此后，史等人 [33] 提出通过像素变换层实现可学习的上采样。例如，VDSR[20] 对低分辨率图像进行上采样，以达到目标分辨率，然后使用超分辨率模型还原局部细节和纹理，因此可以将该方法看做缩放因子任意的超分辨率模型。然而，该方法的明显缺点是：输出图像尺寸增大，计算量增大。因此，后续的方法专门解决缩放因子固定且为整数的问题 (如：放大 2 倍或 4 倍)。

最近，胡等人 [15] 提出使用任意尺度缩放模块代替先前方法的缩放倍数固定的上采样层。Meta-SR[15] 使用动态卷积核解决放大倍数为实数的问题。随后，王等人 [37] 介绍了尺度估计特征和上采样模块，以重建得到目标分辨率任意的图像。先前的方法主要在水平轴或竖直轴使用超分辨率模型。然而，我们在 SRWarp 中提出的扭曲模块与之不同，它允许图像变换成任意形状。

卷积神经网络中的不规则空间采样 在数字图像中，像素均匀排列在二维矩形中。然而，物体在图像中可能以任意形状和方向出现，因此使用简单的卷积难以解决这一问题。为克服这一局限，空间 Transformer 网络 [16] 能够估计合适的扭曲参数，以补偿输入图像中可能的变形。通过可变形卷积 [6] 和感受野形状不定的卷积 [17] 预测依赖于输入图像的卷积核偏移量和调节器 [54]，以实现不规则空间采样。此外，可变形卷积方法 [10] 能够重置卷积核的权重，自适应地改变有效的感受野。最新的图像复原模型 (特别是时序数据)，引入不规则采样策略，以实现准确对齐 [36, 38, 43]。然而，我们的方法首次尝试将图像扭曲问题看做超分辨

率问题的。

3 方法

我们将广义超分辨率框架命名为 SRWarp。$I_{LR} \in \mathbb{R}^{H\times W}, I_{HR} \in \mathbb{R}^{H'\times W'}$ 和 $I_{SR} \in \mathbb{R}^{H'\times W'}$ 分别表示低分辨率图像，参考高分辨率图像和通过模型预测的目标超分辨率图像。$H \times W$ 和 $H' \times W'$ 指图像分辨率，并且我们忽略 RGB 色彩通道以简化问题。和传统的超分辨率模型不同，目标分辨率 $H' \times W'$ 取决于所使用的变换。由于扭曲可能产生非矩形的不规则的形状，因此我们用图像的边界框定义分辨率。若需获取此部分的详细描述和分析，请参考我们的补充材料。

3.1 单应性变换下的超分辨率

给定 3×3 的单应性矩阵 M 和源图中的一点 $p = (x, y, 1)^T$，由于 $Mp = p'$ 或 $f_M(x, y) = (x', y')$(f_M 是相应的函数表示)，我们可以计算目标单应坐标 $p' = w'(x', y', 1)^T$。在逆向扭曲中，通过 $p = M^{-1}p'$ 计算每个输出像素 p' 在源图中的对应位置。如果我们仅沿 x 方向和 y 方向缩放图像，那么单应性矩阵 M 被定义为：

$$M_{s_x s_y} = \begin{pmatrix} s_x & 0 & 0.5(s_x - 1) \\ 0 & s_y & 0.5(s_y - 1) \\ 0 & 0 & 1 \end{pmatrix} \tag{1}$$

s_x 和 s_y 分别表示沿 x 方向和 y 方向的缩放因子，变换部分 $0.5(s_* - 1)$ 通过亚像素位移保证准确对齐 [37]。早期大多数超分辨率模型用于解决 $s_x = s_y$ 的问题 (s_x 和 s_y 是预先定义的整数 [33, 25, 12, 39])。最近的方法放宽了这种限制，允许 s_x 和 s_y 是任意实数 [15, 37]。然而，单应性矩阵 M 仍存在大量可能的形式。

图 2从图像尺度金字塔的角度展示了传统超分辨方法和我们的 SRWarp 方法。为简便起见，我们假设低分辨率图像 I_{LR} 位于 $z = 1$ 的平面，放大 s 倍的超分辨率结果 I_{SR} 相当于金字塔 $z = s$ 的截面，此时超分辨率图像中所有点的 z 坐标相同。先前的方法旨在学习平行于所给低分辨率图像的图像表示。然而，图像金字塔中的任意截面，或空间中的一半平面都可以作为超分辨率结果。因此，我们将扭曲问题重定义为空间尺度、长宽比变化的广义超分辨率模型，因为结果图像中的像素可以根据它们的位置有不同的 z 坐标。

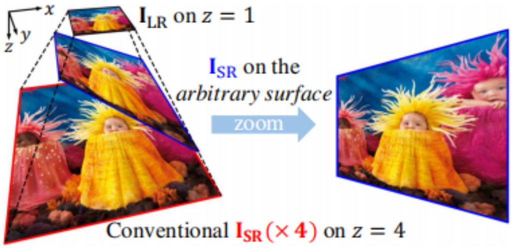

x
y
z
$\mathbf{I}_{\mathrm{LR}}$ on $z = 1$
$\mathbf{I}_{\mathrm{SR}}$ on the *arbitrary surface*
zoom
Conventional $\mathbf{I}_{\mathrm{SR}}(\times 4)$ on $z = 4$

图 2: **广义超分辨率的概念。**虚线表示在 xyz 坐标系中的采样图像金字塔。红色图像是使用传统方法得到的放大 s 倍的超分辨率图像，它只在 $z = s$ 的平面；而蓝色图像是我们模型得到的结果，它能够存在于金字塔的任何一个切面。并且，我们指出金字塔的长宽比也可以变化。

3.2 自适应翘曲层

图像扭曲包括两种基本操作，分别是映射和重采样。映射初步确定输入和输出图像的空间关系。若目标坐标是 $p' = (x', y')$，对应源图像中的像素坐标为 $p = (x, y) = f_M^{-1}(x', y')$。为简便起见，我们忽略了齐次表示。然而数字图像中的像素只存在于整数坐标，x 和 y 可能是任意实数，这取决于映射函数 f_M^{-1}。因此，需要根据以下公式进行合适的重采样才能获得合理的像素值。

$$\boldsymbol{W}(x', y') = \sum_{i,j=a}^{b} \boldsymbol{k}(x', y', i, j)\boldsymbol{F}(\lfloor x \rceil + i, \lfloor y \rceil + j) \tag{2}$$

其中，$\lfloor \cdot \rceil$ 是取四舍五入运算符，$\boldsymbol{F} \in \mathbb{R}^{H\times W}$ 是输入，$\boldsymbol{W} \in \mathbb{R}^{H'\times W'}$ 是输出，k 是点插值核。a 和 b 是 $k \times k$ 窗口的边界索引，且 $k = b - a + 1$。例如，对于标准的 3×3 的核，我们可以设置 $a = -1, b = 1$。

传统的重采样算法使用固定的采样坐标和核函数计算权重 k，不考虑变换 M。例如，如图 3a 所示，广泛使用的双三次翘曲首先计算 $k \times k$ 窗口中每个点相对于 (x, y) 的相对偏移量 (o_x, o_y)，然后使用三次样条构造 k。然而，由于变换的多样性，这一公式在一些方面并非最优。首先，当目标图像未定义在矩形网格时，变换后的几何形状难以考虑。其次，固定的核函数

限制了可推广性，而最近的超分辨率模型更倾向于可学习的上采样 [33, 15, 37]，而不是固定缩放倍数的模型 [20]。为解决这些问题，我们提出了自适应翘曲层 (AWL)，使重采样核 k 可以考虑局部变形。

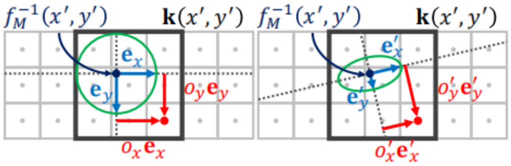
为对每个目标点 (x', y') 确定合适的采样坐标，我们使用雅克比矩阵 $J(x', y') = (\boldsymbol{u}^T \ \boldsymbol{v}^T)$ 对反向映射 M^{-1} 线性化。$\boldsymbol{u}$ 和 $\boldsymbol{v}$ 的计算公式如下：

$$\begin{aligned} \boldsymbol{u} &= \frac{f_M^{-1}(x' + \epsilon, y') - f_M^{-1}(x' - \epsilon, y')}{2\epsilon} \\ \boldsymbol{v} &= \frac{f_M^{-1}(x', y' + \epsilon) - f_M^{-1}(x', y' - \epsilon)}{2\epsilon} \end{aligned} \tag{3}$$

其中，$f_M^{-1} = f_{M^{-1}}$，且 $\epsilon = 0.5$。通过局部近似的方法，源图像上以点 (x, y) 圆心的的单位圆在目标平面被映射为椭圆 [11]。然后，我们计算了椭圆的两个主轴 e'_x 和 e'_y。如图 3b 所示，相对位移向量 $\boldsymbol{o} = (o_x, o_y)$ 在新的局部自适应坐标系中被表示为 $\boldsymbol{o'} = (o'_x, o'_y)$。在重采样过程中，每个点的实际贡献是根据相对于原点的距离计算的。因此，考虑到局部畸变，我们利用自适应坐标调节点。原始的位移向量被缩放为 $\frac{||o'||}{||o||}\boldsymbol{o}$，并以此计算重缩放核 k。

然后，我们引入了核估计量 $\mathcal{K}$ 以估计自适应重采样权重 $\boldsymbol{k}$。与传统的插值方法类似，它需要 k^2 个偏移向量来确定窗口 $\boldsymbol{k}(x', y')$ 内每点 $\boldsymbol{F}(x, y)$ 的贡献。然而，我们采用了一系列的全连接层 [15, 37] 以学习这一功能，而不是使用预先确定的重采样核。可学习的网络允许考虑局部变形，并且为给定的变换生成合适的动态卷积核 [18, 19]。我们通过结合自适应重采样网格和核预测层 $\mathcal{K}$ 以构建自适应翘曲层 $\mathcal{W}$。

$$\boldsymbol{W} = \mathcal{W}(\boldsymbol{F}, \{_{\mathcal{M}})$$



(a) Regular resampling (b) Adaptive resampling

图 3: **自适应网格的示例。**每一点表示源平面 $\boldsymbol{F}$ 的一个杨素。(a) 在规则网格中，重采样基 e_x 和 e_y(蓝色) 正交，且与源图像对齐。(b) 我们为每个目标点 (x', y') 设置不规则重采样基 e'_x 和 e'_y，它们的长度和方向不同。随着重采样核的变化，一个示例点的相对位移向量 (o_x, o_y)(红色) 被映射至 (o'_x, o'_y)。绿色的椭圆展示了目标图像的单位圆如何投影到源平面。

3.3 多尺度混合

图 2说明广义超分辨率任务下的图像存在空间尺度变化下的扭曲。因此，多尺度表示在重建高质量图像中至关重要。为了有效利用这一特性，我们为 SR-Warp 框架引入一种混合方法。

多尺度特征提取器 我们定义了一个特定尺度的特征提取器 $\mathcal{F}_{\times f}$，缩放因子为整数 s[27, 39]。给定一张低分辨率图像 $\boldsymbol{I}_{LR}$，这一模块能够提取尺度特定的特征 $\boldsymbol{F}_{\times s} \in \mathbb{R}^{C \times sH \times sW}$。其中，$C$ 是输出通道数。尽管它可以为每个尺度因子分离网络，但在实际应用中，我们采用的是具有多个上采样层 [27] 的共享特征提取器。例如，先前的方法已经证实多尺度表示可以在一个单一模型中联合学习 [20, 22, 23, 27]。此外，与使用多个模型提取空间特征相比，使用共享骨架网络的计算量更小。如图所示，根据最前沿的超分辨率结构，我们将最后一个上采样层替换为 ×1, ×2 和 ×4 的特征提取器以实现我们的多尺度框架。

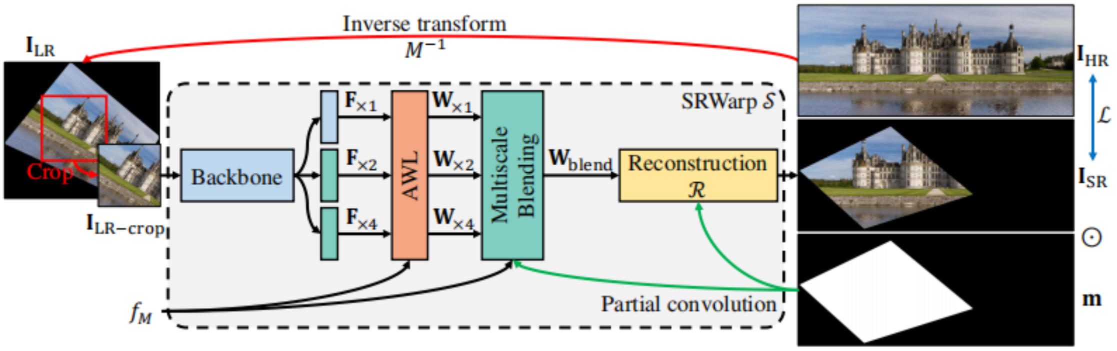


图 4: **SRWarp 模型的整体架构。**详细结构见补充材料。扭曲图像 I_{SR} 外围的黑色区域表示被忽略的空白像素

多尺度扭曲和混合 对于每个特定尺度的特征 $\boldsymbol{F}_{\times s}$，我们构建了相应的 $MM_{s^{-1}s^{-1}}$ 的变换。这一结果表明不同分辨率的特征能够映射到一个固定的空间维度 $H' \times W'$。我们使用 $\boldsymbol{W}_{\times s} \in \mathbb{R}^{C \times H' \times W'}$ 表示扭曲后的特征：

$$\boldsymbol{W}_{\times s} = \mathcal{W}(\mathcal{F}_{\times s}(\boldsymbol{I}_{LR}), MM_{s^{-1}s^{-1}})$$

然后，可以根据一系列多尺度的扭曲特征 $\{\boldsymbol{W}_{\times s} | s = s_0, s_1, \cdots\}$ 重建得到输出的高分辨率图像 $\boldsymbol{I}_{SR}$。然而，这些特征的简单组合 (如平均或关联) 不能够反映广义超分辨率问题的空间变换特性。因此，我们介绍一个多尺度混合模块以有效结合不同分辨率的信息。为确定每个局部区域的合适尺度，图像的内容起至关重要的作用。比如，在翘曲过程中使用低频组件能够防止混叠和不期望的伪影出现。相反，高频率的细节能够准确地表示边缘和纹理。因此，我们使用可学习的特定尺度和全局内容尺度提取器 $\mathcal{C}_{\times s}$ 和 $\mathcal{C}$。

$$\boldsymbol{C} = \mathcal{C}(\mathcal{C}_{\times s_0}(\boldsymbol{W}_{\times s_0}), \mathcal{C}_{\times s_1}(\boldsymbol{W}_{\times s_1}), \cdots)$$

其中，全局内容特征 $\boldsymbol{C} \in \mathbb{R}^{C \times H' \times W'}$ 能够由尺度特定的特征 $\mathcal{C}_{\times s_i}(\boldsymbol{W}_{\times s_i})$ 表示。

由于我们的 SRWarp 方法能够处理空间变化的扭曲，适当的特征尺度同样依赖于局部变化。我们所提出的模型可以通过像素周围的变换程度决定多尺度的贡献率。因此，我们获取的尺度特征 $\boldsymbol{S} \in \mathbb{R}^{H' \times W'}$ 为：

$$S(x', y') = -\log|\det(J(x', y'))|$$

物理上，雅可比矩阵的行列式描述了变换的局部放大因子。当采用反向映射时，我们考虑雅可比行列式的倒数，并通过取自然对数进行标准化。

我们的多尺度混合模块在关联内容和尺度特征 $\boldsymbol{C}$ 和 $\boldsymbol{S}$ 中使用 1×1 卷积。通过这种方法，可以为每个输出坐标 (x', y') 确定合适的混合权重 $w_{\times s}$。混合特征 $\boldsymbol{W}_{\text{blend}}$ 可以表示为：

$$\boldsymbol{W}_{\text{blend}} = \sum_s w_{\times s} \odot \boldsymbol{W}_{\times s}$$

其中，$\odot$ 表示元素乘法。

部分卷积 当点映射到源图像外部时，图像扭曲会在目标坐标上产生空白像素。这些区域可能会对模型性能不利，因为传统的卷积神经网络对各像素的考虑程度相同。为有效解决这一问题，我们定义了二维二进制掩码如下：

$$\boldsymbol{m}(x', y') = \begin{cases} 0, \text{ if}(x, y)\text{is outside of } \boldsymbol{F}_{\times 1} \\ 1, \text{ otherwise} \end{cases}$$

其中，$f_M(x, y) = (x', y')$。我们通过 $\times 1$ 特征 $\boldsymbol{F}_{\times 1}$ 计算掩码 $\boldsymbol{m}$，并跨尺度共享以保证不同分辨率的一致性。然后，我们使用局部卷积 [28, 29] 对内容特征提取器 $\mathcal{C}$ 和 $\mathcal{C}_{\times s}$ 忽略空白像素。

3.4 SRWarp

最后，我们引入带有 5 个残差块 [25, 27] 的重建模块 $\mathcal{R}$。如图 4所示，我们将超分辨率骨架、自适应翘曲层和重建模块结合，构造出 SRWarp 模型 $\mathcal{S}$。对于稳定训练，残差连接 $\boldsymbol{I}_{\text{SR}} = \mathcal{R}(\boldsymbol{W}_{\text{blend}}) + \boldsymbol{I}_{\text{bic}}$。其

中，$\boldsymbol{I}_{\text{bic}}$ 是使用双三次插值得到的扭曲图像，$\boldsymbol{I}_{\text{SR}}$ 是最终的输出。给定一组训练输入和输出 $(\boldsymbol{I}^n_{\text{LR}}, \boldsymbol{I}^n_{\text{HR}})$，我们对重建图像和 ground-truth 的平均 L_1[22, 27] 误差最小化：

$$\mathcal{L} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{||m||_0}||m \odot (\mathcal{S}(\boldsymbol{I}^n_{\text{LR}}, f_M) - \boldsymbol{I}^n_{\text{HR}})||_1$$

其中，$N = 4$ 是一批样本的数量，n 是样本下标，0 范数表示非零值的数量，$\mathcal{S}(\boldsymbol{I}^n_{\text{LR}}, f_M) + \boldsymbol{I}_{\text{bic}} = \boldsymbol{I}_{\text{SR}}$。变换函数 f_M 在一批样本中共享以实现高效计算。二进制掩码 $\boldsymbol{m}$[9] 防止空白像素梯度的反向传播。所提出的 SRWarp 模型可以用 ADAM 优化器 [21] 进行端到端训练。

4 实验

我们使用两种不同的超分辨率网络作为 SR-Warp 模型中多尺度特征提取器的主干网络。改进的 MDSR 结构 [27] 作为较小的基线，而具有多尺度分支 (MRDB) 的 RRDB[39] 为提高性能提供了一个更大的主干。更多训练设置见补充材料。PyTorch 代码和数据集的开源地址如下：https://github.com/sanghyun-son/srwarp

4.1 数据集和评价指标

数据集 在传统的图像超分辨方法中，由于许多现实因素 (r 如户外场景的多样性和亚像素未对齐 [4, 5, 50])，难以获取现实世界的低分辨率和高分辨率图像对。与之类似，对于我们的广义 SR 任务，在野外收集具有相应变换矩阵的高质量图像对也很困难。因此，我们从现有的 DIV2K 数据集 [1] 中合成低分辨率样本，提出 DIV2K-Warping(DIV2KW) 数据集，并在该数据集上以监督学习的方式训练 SRWarp 模型。我们首先为训练集、验证集和测试集分配了 500, 100,100 个随机翘曲参数。每个矩阵都包括随机上采样、剪切、选择和映射，因为我们主要希望增大数据集。

在学习阶段，我们随机从 DIV2K 训练集中选取 800 张图像制作高分辨率图像块，并使用翘曲矩阵 M_I^{-1} 构造 ground-truth 图像块 $\boldsymbol{I}_{HR}$。然后，我们通过 M_I^{-1} 获取对应的低分辨率输入 $\boldsymbol{I}_{IR}$。为简便起见，我们将输入中的最大有效方块区域裁剪出来，作为 $\boldsymbol{I}_{IR-crop}$。使用过转换矩阵 M_i 和低分辨率图像块 $\boldsymbol{I}_{IR-crop}$，我们通过使扭曲模型最优，以重建高分辨率图像为 $\boldsymbol{I}_{HR}$。图 4展示了 SRWarp 模型的实际训练过程。我们对 DIV2K 验证集的 100 张图像进行不同的变换，使用同样的方式评估模型。

评价指标 我们在 RGB 色彩空间使用传统的 PSNR 指标评价扭曲图像的质量。然而，与训练目标类似，我们只考虑了 $H' \times W'$ 网格中的有效像素，因为它们的形状不规则，不是标准的矩形。修改后的 PNSR 指标中含有二进制掩码 $\boldsymbol{m}$(mPSNR)：

$$\text{mPSNR(dB)} = 10\log_{10}\frac{||m||_0}{||m \odot (I_{SR} - I_{HR})||_2^2} \tag{4}$$

其中，图像 I_* 被标准化到 $0 \sim 1$ 的范围。

4.2 消融实验

如表 1 所示，由于各模块互不影响，我们对其进行组合验证。我们使用修改后的 MDSR 基线结构作为轻量级测评的主干网络。我们将单尺度超分辨率主干网络和含有标准翘曲层的模型结合，作为基线模型。A, M 和 R 表示对应的构型。如：A-R 模型的 mPSNR 值为 32.19dB。我们的 SRWarp 模型可以用 A-M-R 表示，mPSNR 值达到 32.29dB。

自适应翘曲层 如表 1所示，AWL 通过提供空间自适应的重采样核，提高了模型性能。表 2a 对比了 SR-Warp 方法中 AWL 的可能策略。如图 3b 所示，我们用空间变换的表示 (自适应策略如表 2a 所示) 代替了 M-R 常规的重采样网格。然而，由于重采样权重 $\boldsymbol{k}$ 不可学习，即使考虑到空间变化的性质，该公式也没有带来性能优势。在常规网格中引入核估计器 (该层如表 2b 所示) 能够在 M-R 方法上获得 +0.06dB 的 mPNSR 增益。通过结合空间变换坐标和可学习模块，性能进一步提高到 32.29dB。我们注意到，自适应重采样网络在每个输出的位置不需要额外的参数，这使得计算更加简便。

表 1: **SRWarp 方法中每个模块的贡献**。A, M, R 和 B 分别表示自适应翘曲层 (AWL)、多尺度混合模块、重建模块和主干框架。括号中的数字表示超过基线模型的性能增益

A	M	R	B	DIV2KW$_{Valid}$ 的 mPSNR
-	-	-	EDSR	31.36(+0.00)
✓	-	-		32.06(+0.70)
-	-	✓		32.08(+0.72)
✓	-	✓		32.19(+0.83)
-	✓	-	MDSR	32.19(+0.83)
✓	✓	-		32.21(+0.85)
-	✓	✓		32.19(+0.83)
✓	✓	✓		**32.29(+0.93)**
-	-	-	RRDB	31.64(+0.28)
✓	✓	✓	MRDB	32.56(+1.20)

我们分析了 AWL 的两种可能的变体。如表 2a 所示，AWL-SS 跨尺度和通道维度共享核估计器 $\mathcal{K}$，即使是在多尺度超分辨率主干模型也是如此。如表 2a 所示，AWL-MS 通过使用尺度特定模块 $\mathcal{K}_{\times s}$，获得了较小的性能增益。在所提出的 SRWarp 模型中，我们以另一种方式进一步估计核 (每个点 (x', y') 的权重数量为 $C \times k \times k$)，mPSNR 的性能增益是 +0.05dB。

表 2: **自适应翘曲层和多尺度混合策略在 SRWarp 模型中的作用**。我们在 DIV2KW$_{Valid}$ 数据集上评估了每种方法。在相同的环境下，SRWarp 的 mPSNR 达到了 32.29dB。

Method	mPSNR
Adaptive	32.19
Layer	32.25
AWL-SS	32.23
AWL-MS	32.24

Method	mPSNR
Average	32.26
Concat.	32.29
w/o C	32.24
w/o S	32.23

多尺度混合 如表 1所示，我们的多尺度方法 (M) 与 EDSR[27] 基线单尺度模型相比 (表 1中的 A-R)，极大地提高了空间变换超分辨率模型的性能。我们对表 2b 中的混合模块进行修改，改变计算组合系数 $w_{\times s}$ 的公式。有趣的是，仅使用对翘曲特征 $\boldsymbol{W}_{\times s}$ 求平均的方式会比使用 1×1 卷积 (表 2b) 进行连接和混合的方式效果好。性能的下降表明，有效的混合模块需要适当的设计，因为连接是一个更通用的公式。同样，我们分析了内容和尺度特征是怎样使混合模块将多尺度表示有效结合的。如果忽略内容信息 (表 2b 中的 w/o C)，我们的 SRWarp 模型的 mPSNR 指标会降低 0.05dB。移除尺度特征模块 (表 2b 中的 w/o S) 同样会带来类似的性能下降，这证实了我们所提出的多尺度混合是有效的。

重建模块和部分卷积 因为重建模块 $\mathcal{R}$ 能进一步调整输出图像，包含该模块 (表 1中的 *-R) 的模型都能过获得额外的性能增益。我们同样检查了部分卷积 [28, 29] 在内容特征提取器和重建模块的必要性。和先前的超分辨率方法相比，由于某些因素，我们的 SRWarp 框架对边界效应更加敏感。首先，图像边界 (如：有效区域和空白区域之间的部分) 无法与卷积核对齐，形状不规则。其次，因为我们在常规的二维网格中放置了不规则形状的数据，翘曲图像中大量的空白像素会对后续的可学习层造成负面影响。在没有部分卷积的情况下，SRWarp 的收敛速度更慢，且由于严重的边界效应，最终性能下降了 0.06dB。

主干网络 表 1的后两行展示了使用不同主干网络时 SRWarp 模型的性能。与 MDSR 网络 (参数量为 17.1M) 相比，使用更大的 MRDB 网络 (参数量为 17.1M) 时，会获得 0.27dB 的 PSNR 提升。这表示，使用 MRDB 作为主干网络，训练数据的拟合度更好，验证集的性能更好。

4.3 与其他方法的对比

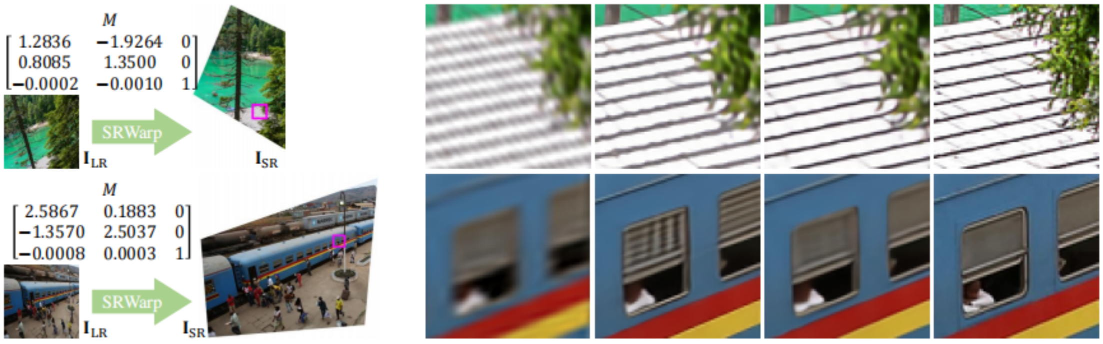
我们用 SRWarp 和现有的其他模型进行对比。我们注意到，由于我们的方法是第一次尝试广义图像 SR，很难与其他方法进行精确的比较。首先，我们采用了 OpenCV[3] 中传统的基于插值的扭曲算法。我们使用带有双三次插值核的 cv2.WarpPerspective 函数合成

扭曲的图像。对于替代方案，我们结合了最先进的超分辨率模型和传统的翘曲操作。由于给定的低分辨率图像在插值前被过采样，翘曲函数可以直接合成高质量的结果。我们注意到转换矩阵 M 被 $MM_{s^{-1}s^{-1}}$ 补偿，因为超分辨率模型的输出是原始输入的 $\times s$ 倍。

表 3显示了多种方法的量化比较。为公平起见，我们采用了 DIV2KW$_{Test}$ 数据集，而没有使用 4.2 节中的验证集。和传统的 cv2 算法相比，使用超分辨率方法能够使 mPSNR 至少提高 2.41dB。高倍模型 (×4) 比 ×2 模型的效果更好，这验证了细粒度上采样的重要性。我们的 SRWarp 模型与其他基于超分辨率的模型相比，性能有更大的提升。如图 5所示，我们的方法能够重建出锐度更高、混叠更少的图像，这说明了 AWL 和多尺度混合层的有效性。

表 3: SRWarp 与其他翘曲算法的对比。+cv2 表示我们首先应用特定尺度的 SR 模型进行上采样，然后用传统的翘曲算法对上采样后的图像进行变换。括号中的数字表示与 ×2RDN+cv2 方法相比的性能增益。最优和次优性能分别用**加粗**和<u>下划线</u> 标注。

Method			DIV2KW$_{Test}$ 的 mPSNR
cv2(Bicubic)[3]			27.85(-2.41)
×2	RDN[52]	+cv2	30.22(+0.00)
	EDSR[27]		30.42(+0.20)
	RCAN		30.45(+0.23)
×4	RDN[52]	+cv2	30.50(+0.28)
	EDSR[27]		30.66(+0.44)
	RCAN		30.71(+0.49)
	RRDB[39]		<u>30.76(+0.54)</u>
SRWarp(MRDB)			**31.04(+0.82)**



(a) $\mathbf{I}_{\mathrm{LR}}$, $\mathbf{I}_{\mathrm{SR}}$, and M (b) cv2 (c) RRDB + cv2 (d) **SRWarp** (e) GT

图 5: **DIV2KW$_{Test}$ 数据集的定性扭曲结果**。我们使用扭曲矩阵 M_i 构造输入 LR 和输出 HR 图像。为简便起见，我们省略了变换组件。图像块由 DIV2KW$_{Test}$ 数据集中的 0.807.png 和 0.850.png 裁剪得到。

4.4 任意尺度的超分辨率模型

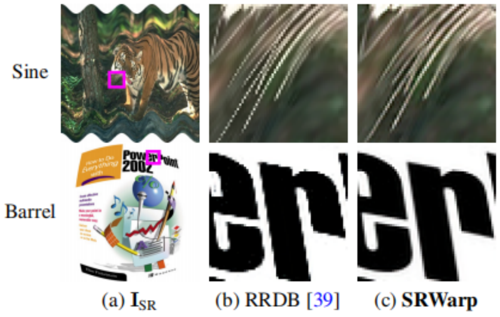
我们的 SRWarp 模型对传统的使用尺度变换矩阵的超分辨率模型进行推广。为了证明所提出的框架与现有的公式相兼容，我们在任意尺度因子的常规 SR 任务上评估了我们的方法。我们使用 RRDB 主干网络 [39] 在分数级的 DIV2K 数据集 [15] 上进行训练，并根据胡等人的方法进行评估 [15]。SRWarp 模型的输入变换受 (1) 的限制，其他的配置是固定的。表 4 展示了 B100 数据集 [31] 上超分辨率结果与 ground-truth 相比的 Y 通道平均 PSNR 指标。我们注意到 SRCNN[8] 和 VDSR[20] 先将图片缩放至任意分辨率，再将其输入至网络。与元上采样模块 (Meta-EDSR 和 Meta-RDN) 相比，我们的自适应翘曲层和多尺度混合模块能够有效地使模型应用于任意分辨率的超分辨率模型。

表 4: **任意缩放尺度的超分辨率任务定量比较。**我们使用每种方法的官方代码实现，并在统一的环境中进行比较。在 ×3.0 超分辨率任务上使用 100 张测试图像进行评估，不包括初始化、I/O 和其他开销。对于 SRCNN[8] 模型，×3 的网络 (9-5-5) 是通过所有缩放因子在 CPU 上进行评估的。我们的 SRWarp 模型的参数量少于其他模型 (Meta-EDSR 和 Meta-RDN)，但性能总是优于其他方法。

			B100 数据集任意缩放尺度的 PSNR								
Method	#Params	Runtime	×2.0	×2.2	×2.5	×2.8	×3.0	×3.2	×3.5	×3.8	×4.0
SRCNN[8]	0.06M	2340ms	27.11	27.85	28.62	28.71	28.37	27.89	27.17	26.59	26.27
VDSR[20]	0.67M	26ms	31.82	30.36	29.54	28.84	28.77	28.15	27.82	27.46	27.27
Meta-EDSR[15]	40.1M	218ms	32.26	31.31	30.40	29.61	29.22	28.82	28.27	27.86	27.67
Meta-RDN[15]	22.4M	253ms	**32.33**	31.45	**30.46**	29.69	29.26	28.88	28.41	**28.01**	27.71
SRWarp(MRDB)	18.3M	155ms	32.31	**31.46**	**30.46**	**29.71**	**29.27**	**28.89**	**28.42**	**28.01**	**27.77**

4.5 除单应变换外的其他变换

我们的模型仅仅关注了单应变换。然而，我们可以对方法进行扩展，将其扩展至任意的反向映射 $f_M^{-1}(x', y') = (x, y)$。虽然我们只对训练采用单应变换，但自适应翘曲层和多尺度混合能在其他未知变换上得到很好的推广。图 6将我们的方法和将 RRDB、传统双三次差值结合的方法进行对比。我们的 SRWarp 模型在一般的图像编辑任务中更加灵活多样，能够重建得到视觉效果较好的结果。



(a) $\mathbf{I}_{\text{SR}}$ (b) RRDB [39] (c) **SRWarp**

图 6: **使用 SRWarp 的图像扭曲超分辨效果。**我们对 B100 数据集 [31] 中的 108005.png 和 Set14 数据集 [46] 中的 ppt3.png 进行多种变换。(b)RRDB 对应于表 3中的 RRDB+cv2。

5 总结

我们首次在图像变换下对传统 SR 任务进行推广。我们的 SRWarp 框架能够处理空间变化的上采样任务，输出图像的分辨率和形状任意。我们还通过消融实验验证了各个新组件的贡献，如：自适应翘曲层和多尺度混合层。通过比较图像的效果说明了需要通过 SRWarp 模型进行图像扭曲变换的原因，验证了该方法的优势。

5.1 致谢

本研究得到韩国科学和通信技术部 IITP 资助 (No. 2017-0-01780) 以及和现代汽车公司和起亚汽车公司通过 HKMC-SNUAI 联盟基金资助的 IITP 资助。

参考文献

[1] Eirikur Agustsson and Radu Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In CVPR Workshops, 2017.

[2] Yuval Bahat and Tomer Michaeli. Explorable super resolution. In CVPR, 2020.

[3] G. Bradski. The OpenCV Library. Dr. Dobb’s Journal of Software Tools, 2000.

[4] Jianrui Cai, Hui Zeng, Hongwei Yong, Zisheng Cao, and Lei Zhang. Toward real-world single image super-resolution: A new benchmark and a new model. In ICCV, 2019.

[5] Chang Chen, Zhiwei Xiong, Xinmei Tian, Zheng-Jun Zha, and Feng Wu. Camera lens super-resolution. In CVPR, 2019.

[6] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In ICCV, 2017.

[7] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In CVPR, 2019.

[8] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. TPAMI, 2016.

[9] Ying Fu, Tao Zhang, Yinqiang Zheng, Debing Zhang, and Hua Huang. Hyperspectral image super-resolution with optimized RGB guidance. In CVPR, 2019.

[10] Hang Gao, Xizhou Zhu, Steve Lin, and Jifeng Dai. Deformable kernels: Adapting effective receptive fields for object deformation. In ICLR, 2020.

[11] N. Greene and P. S. Heckbert. Creating raster omnimax images from multiple perspective views using the elliptical weighted average filter. CG & A, 6(6):21–27, 1986.

[12] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Deep back-projection networks for super-resolution. In CVPR, 2018.

[13] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita. Space-time-aware multi-resolution video enhancement. In CVPR, 2020.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 2016.

[15] Xuecai Hu, Haoyuan Mu, Xiangyu Zhang, Zilei Wang, Tieniu Tan, and Jian Sun. Meta-SR: A magnification-arbitrary network for super-resolution. In CVPR, 2019.

[16] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In NIPS, 2015.

[17] Yunho Jeon and Junmo Kim. Active convolution: Learning the shape of convolution for image classification. In CVPR, 2017.

[18] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. In NIPS, 2016.

[19] Younghyun Jo, Seoung Wug Oh, Jaeyeon Kang, and Seon Joo Kim. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In CVPR, 2018.

[20] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In CVPR, 2016.

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In ICLR, 2015.

[22] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and MingHsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In CVPR, 2017.

[23] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and MingHsuan Yang. Fast and accurate image super-resolution with deep laplacian pyramid networks. TPAMI, 41(11):2599–2613, 2018.

[24] Hoang Le, Feng Liu, Shu Zhang, and Aseem Agarwala. Deep homography estimation for dynamic scenes. In CVPR, 2020.

[25] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In CVPR, 2017.

[26] Wenbo Li, Xin Tao, Taian Guo, Lu Qi, Jiangbo Lu, and Jiaya Jia. MuCAN: Multi-correspondence aggregation network for video super-resolution. In ECCV, 2020.

[27] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In CVPR Workshops, 2017.

[28] Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In ECCV, 2018.

[29] Guilin Liu, Kevin J. Shih, Ting-Chun Wang, Fitsum A. Reda, Karan Sapra, Zhiding Yu, Andrew Tao, and Bryan Catanzaro. Partial convolution based padding. In arXiv, 2018.

[30] Andreas Lugmayr, Martin Danelljan, Luc Van Gool, and Radu Timofte. SRFlow: Learning the super-resolution space with normalizing flow. In ECCV, 2020.

[31] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In ICCV, 2001.

[32] Ben Niu, Weilei Wen, Wenqi Ren, Xiangde Zhang, Lianping Yang, Shuzhen Wang, Kaihao Zhang, Xiaochun Cao, and Haifeng Shen. Single image super-resolution via a holistic attention network. In ECCV, 2020.

[33] Wenzhe Shi, Jose Caballero, Ferenc Huszar, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In CVPR, 2016.

[34] Rahul Swaminathan and Shree K Nayar. Non-metric calibration of wide-angle lenses and poly-cameras. TPAMI, 22(10):1172–1178, 2000.

[35] Yang Tan, Haitian Zheng, Yinheng Zhu, Xiaoyun Yuan, Xing Lin, David Brady, and Lu Fang. CrossNet++: Cross-scale large-parallax warping for reference-based super-resolution. TPAMI, 2020.

[36] Yapeng Tian, Yulun Zhang, Yun Fu, and Chenliang Xu. TDAN: Temporally-deformable alignment network for video super-resolution. In CVPR, 2020.

[37] Longguang Wang, Yingqian Wang, Zaiping Lin Lin, Jungang Yang, Wei An, and Yulan Guo. Learning for scale arbitrary super-resolution from scale-specific networks. arXiv, 2020.

[38] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. EDVR: Video restoration with enhanced deformable convolutional networks. In CVPRW, 2019.

[39] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. ESRGAN: enhanced super-resolution generative adversarial networks. In ECCV Workshops, 2018.

[40] Yingqian Wang, Longguang Wang, Jungang Yang, Wei An, Jingyi Yu, and Yulan Guo. Spatial-angular interaction for light field image super-resolution. In ECCV, 2020.

[41] Pengxu Wei, Ziwei Xie, Hannan Lu, ZongYuan Zhan, Qixiang Ye, Wangmeng Zuo, and Liang Lin. Component divide-and-conquer for real-world image super-resolution. In ECCV, 2020.

[42] Bartlomiej Wronski, Ignacio Garcia-Dorado, Manfred Ernst, Damien Kelly, Michael Krainin, Chia-Kai Liang, Marc Levoy, and Peyman Milanfar. Handheld multi-frame super-resolution. ACM TOG, 38(4):1–18, 2019.

[43] Xiaoyu Xiang, Yapeng Tian, Yulun Zhang, Yun Fu, Jan P. Allebach, and Chenliang Xu. Zooming Slow-Mo: Fast and accurate one-stage space-time video super-resolution. In CVPR, 2020.

[44] Zhucun Xue, Nan Xue, Gui-Song Xia, and Weiming Shen. Learning to calibrate straight lines for fisheye image rectification. In CVPR, 2019.

[45] Jing Yao, Danfeng Hong, Jocelyn Chanussot, Deyu Meng, Xiaoxiang Zhu, and Zongben Xu. Cross-attention in coupled unmixing nets for unsupervised hyperspectral super-resolution. In ECCV, 2020.

[46] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In Curves and Surfaces, 2010.

[47] Jirong Zhang, Chuan Wang, Shuaicheng Liu, Lanpeng Jia, Nianjin Ye, Jue Wang, Ji Zhou, and Jian Sun. Content-aware unsupervised deep homography estimation. In ECCV, 2020.

[48] Lei Zhang, Jiangtao Nie, Wei Wei, Yanning Zhang, Shengcai Liao, and Ling Shao. Unsupervised adaptation learning for hyperspectral imagery super-resolution. In CVPR, 2020.

[49] Shuo Zhang, Youfang Lin, and Hao Sheng. Residual networks for light field image super-resolution. In CVPR, 2019.

[50] Shuo Zhang, Youfang Lin, and Hao Sheng. Residual networks for light field image super-resolution. In CVPR, 2019.

[51] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In ECCV, 2018.

[52] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In CVPR, 2018.

[53] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image restoration. TPAMI, 2020.

[54] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable ConvNets v2: More deformable, better results. In CVPR, 2019.