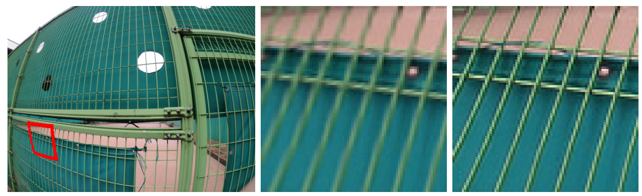
This CVPR 2021 paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

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.



(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.

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].

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

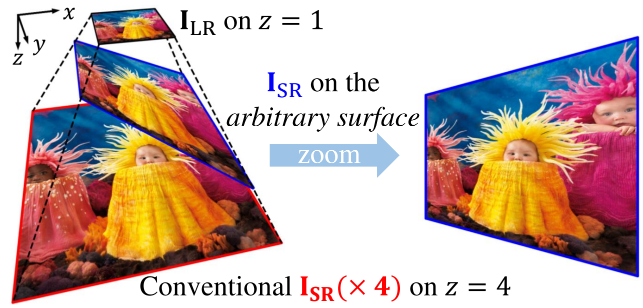


Figure 2: **Concept of the generalized SR.** Dotted lines represent the upscale image pyramid in xyz-coordinate. While $\times s$ super-resolved images from conventional methods (**red**) exist on the plane $z = s$ only, our results (**blue**) exists on any arbitrary cutting surfaces of the pyramid.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

the function f_M^{-1}. Therefore, an appropriate resampling is required to obtain a plausible pixel value as follows:

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

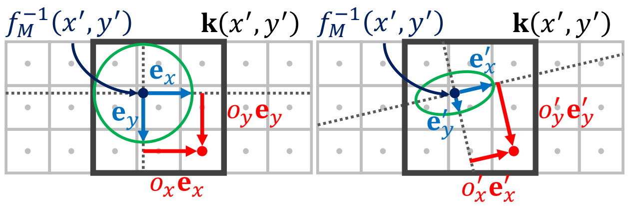
where $\lfloor \cdot \rceil$ is a rounding operator, $\mathbf{F} \in \mathbb{R}^{H \times W}$ is an input, $\mathbf{W} \in \mathbb{R}^{H' \times W'}$ is an output, and $\mathbf{k}$ is a point-wise interpolation kernel, respectively. a and b are boundary indices of the $k \times k$ window, where $k =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

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

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 $\times 1$, $\times 2$, and $\times 4$ 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:

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

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.

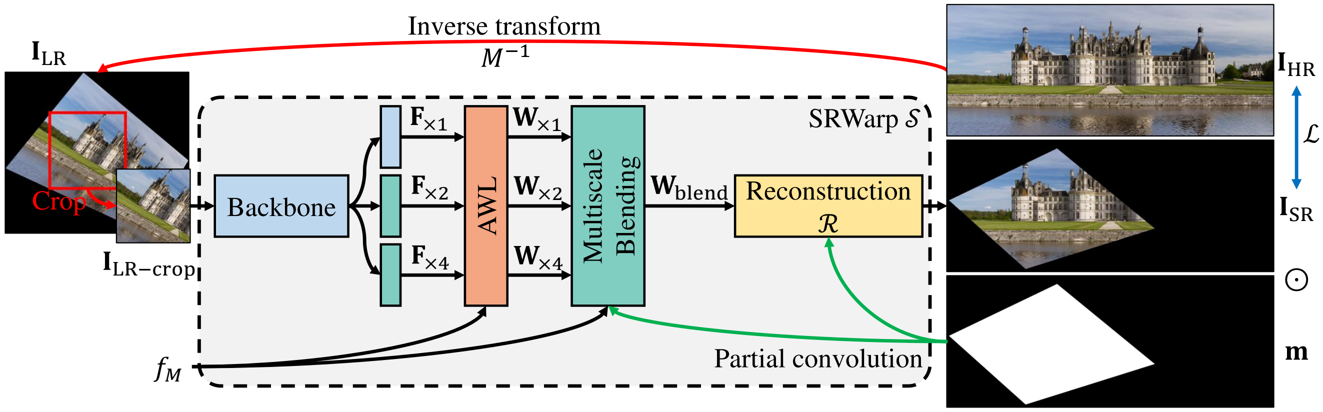


Figure 4: **Overall organization of the proposed SRWarp model.** More detailed architectures are described in our supplementary material. Black regions outside the warped image $\mathbf{I}_{\text{SR}}$ represent void pixels that are ignored.

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:

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

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 $\text{DIV2KW}_{\text{Valid.}}$
–	–	–	EDSR	31.36 (+0.00)
✓	–	–	EDSR	32.06 (+0.70)
–	–	✓	EDSR	32.08 (+0.72)
✓	–	✓	EDSR	32.19 (+0.83)
–	✓	–	MDSR	32.19 (+0.83)
✓	✓	–	MDSR	32.21 (+0.85)
–	✓	✓	MDSR	32.19 (+0.83)
✓	✓	✓	MDSR	**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 $\text{DIV2KW}_{\text{Valid.}}$ 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)
$\times 2$	RDN [52]	+`cv2`	30.22 (+0.00)
	EDSR [27]		30.42 (+0.20)
	RCAN [51]		30.45 (+0.23)
$\times 4$	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 $\times 2$ 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-

Method	# Params	Runtime	PSNR↑(dB) on B100 [31] with arbitrary scale factors								
			×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 $\times 3.0$ SR task using 100 test images excluding initialization, I/O, and the other overheads. For the SRCNN [8] model, the $\times 3$ 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 $\times 4$ models tend to perform better than their $\times 2$ 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.

Sine

Barrel

(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.