Blind Image Super-Resolution with Spatially Variant Degradations

VICTOR CORNILLÈRE, ETH Zurich
ABDELAZIZ DJELOUAH, DisneyResearch|Studios
WANG YIFAN, ETH Zurich
OLGA SORKINE-HORNUNG, ETH Zurich
CHRISTOPHER SCHROERS, DisneyResearch|Studios

Fig. 1. **Upscaling results with spatially varying degradation.** Handling spatially variant degradations is critical when dealing with composited content. In this case the spaceship was composited onto the background image. The two regions have been down scaled with different kernels, and as a result, there is no single kernel that can be used for upscaling the entire image without artifacts. Our method avoids these problems by allowing for automatic local adaptation of the degradation. Photo Credits: Derivative from *Spaceship* by Francois Grassard (CC-BY).

Existing deep learning approaches to single image super-resolution have achieved impressive results but mostly assume a setting with fixed pairs of high resolution and low resolution images. However, to robustly address realistic upscaling scenarios where the relation between high resolution and low resolution images is unknown, blind image super-resolution is required. To this end, we propose a solution that relies on three components: First, we use a degradation aware SR network to synthesize the HR image given a low resolution image *and* the corresponding blur kernel. Second, we train a *kernel discriminator* to analyze the generated high resolution image in order to predict errors present due to providing an incorrect blur kernel to the generator. Finally, we present an optimization procedure that is able to recover both the degradation kernel and the high resolution image by minimizing the error predicted by our kernel discriminator. We also show how to extend our approach to spatially variant degradations that typically arise in visual effects pipelines when compositing content from different sources and how to enable both local and global user interaction in the upscaling process.

CCS Concepts: • **Computing methodologies** → **Image processing**; **Computational photography**.

Additional Key Words and Phrases: Image Super-resolution, Blind Image Super-resolution, Deep Learning

ACM Reference Format:
Victor Cornillère, Abdelaziz Djelouah, Wang Yifan, Olga Sorkine-Hornung, and Christopher Schroers. 2019. Blind Image Super-Resolution with Spatially Variant Degradations. *ACM Trans. Graph.* 38, 6, Article 166 (November 2019), 13 pages. https://doi.org/10.1145/3355089.3356575

Authors' addresses: Victor Cornillère, ETH Zurich, covictor@student.ethz.ch; Abdelaziz Djelouah, DisneyResearch|Studios, abdelaziz.djelouah@disney.com; Wang Yifan, ETH Zurich, yifan.wang@inf.ethz.ch; Olga Sorkine-Hornung, ETH Zurich, olga.sorkine@inf.ethz.ch; Christopher Schroers, DisneyResearch|Studios, christopher.schroers@disney.com.

0730-0301/2019/11-ART166 $15.00
https://doi.org/10.1145/3355089.3356575

1 INTRODUCTION

With the recent advances in deep learning, super-resolution (SR) has become a very active field of research in the past few years. From a practical point of view, obtaining high resolution content from lower quality is beneficial in numerous situations as it allows

to bridge the gap between content resolution and displays. In video production, this offers the possibility to use more affordable cameras while still aiming for 4K content. In addition to this, the *scene-to-screen* workflow, including visual effects, is still largely limited to 2K resolution due to cost and efficiency considerations. A robust and flexible SR solution would offer a faster and cheaper path for producing 4K content.

Initial contributions to the field of learning-based SR have focused on supervised settings with fixed pairs of high/low resolution (HR/LR) images, usually obtained with bicubic downsampling [Dong et al. 2014; Kim et al. 2016; Ledig et al. 2017]. Improvements came primarily from architectural decisions [Kim et al. 2016], training strategies such as adversarial training [Ledig et al. 2017] or a combination of both [Wang et al. 2018]. These solutions achieve impressive results when the downsampling operation is fixed, but result in noticeable artifacts when used on images with a different degradation. If the blur kernel is provided, Shocher et al. [2018] propose a zero-shot SR method that is trained specifically for each image with the given degradation operation, while Zhang et al. [2018d] avoid this per-image training by explicitly providing the kernel to the SR network. Recovering the blur kernel still remains a challenging task, and existing methods rely on priors on image features such as [Michaeli and Irani 2013] that assume image patch redundancy at different scales in the low resolution image.

In this paper, we propose a framework that is able to perform blind SR in a completely automated way while fulfilling the requirements of practical upscaling in video production, such as adaptation to composited content and the possibility of local and global user control. Following the standard approach, we model the low resolution image as a degradation from an "ideal" HR image with blurring and downsampling. First, given a blur kernel and a low resolution image, we train a *degradation-aware generator* network to produce the corresponding high resolution image. A similar approach is used by Zhang *et al.* [2018d]. Second, we observe that providing an incorrect kernel to the generator leads to artifacts in the synthesized image, and we train a *kernel discriminator* network to identify these errors. Instead of estimating the degradation by analyzing the low resolution image, we recover this information by *understanding the artifacts in the high resolution output*. Lastly, we propose an optimization scheme to estimate the degradation parameters that minimize the artifacts in the generator. As a result, we recover both the degradation and the high resolution image. We note that the kernel parameters can be the same in the entire image or locally estimated to deal with cases such as composited content. Our parametrization allows user control for local and global fine-tuning, and our experiments demonstrate the flexibility and robustness of the proposed solution, which is able to handle a large range of downsampling operations. The contributions in this paper are threefold:

- We show how the parameters of the blur kernel can be recovered using a kernel discriminator network to analyze artifacts created by a degradation-aware SR network.
- We propose a framework that leverages the degradation-aware SR network and the kernel discriminator to estimate the blur kernel leading to better SR estimation. The method achieves state-of-the-art results in blind SR.
- An optimization scheme that allows both global and local adaptation of the estimated degradation and SR result.

2 RELATED WORK

The image super-resolution problem is among the earliest problems in image restoration and as such a large number of solutions have been proposed. In this paper we focus on deep learning based methods and SR approaches taking into account the degradation kernel. A detailed review and evaluation of SR state of the art can be found in the survey realized by Nasrollahi and Moeslund [2014] and the benchmark proposed by Yang *et al.* [2014].

In the case of *SR with fixed down-sampling,* deep learning based approaches achieve impressive results by training deep neural networks on pairs of corresponding LR/HR images (or image patches). The first approach to CNN based SR proposed by Dong *et al.* [2014] relies on three steps: patch encoding, non-linear mapping, and reconstruction. Although improvements in terms of quality were achieved by considering deeper network architectures [Dong et al. 2016; Kim et al. 2016], the memory footprint is significant as these methods use a bi-cubic up-sampling of the LR image as input. To avoid the computationally expensive feature extraction in HR, Shi *et al.* [2016] process images in low resolution space and only perform upscaling as a last step. In addition to changes in NN design, Ledig *et al.* [2017] have used generative adversarial networks to achieve improved visual quality. Today a vast amount of work in discriminatively trained SR neural networks exist. Among the noticeable improvements we note the progressive adversarial training approach proposed by Wang *et al.* [2018]. Here a single pyramidal architecture up-samples images to multiple scaling factors, with larger scales benefiting from feature already extracted at lower resolution. In the context of video super-resolution, a standard approach is to rely on consecutive frames to achieve better results [Caballero et al. 2017; Sajjadi et al. 2018]. If motion blur is present, Zhang *et al.* [2018a] propose to jointly solve the deblurring and upscaling problems. Deblurring result is estimated at low resolution and a gate module is used to merge the features extracted from this deblurred result before predicting the high resolution image.

In a different direction, some works have considered *deep regularization priors*. Estimating the high resolution image is expressed as solving a Maximum A Posteriori (MAP) problem. The objective function consists of a fidelity term and a regularization term. Using variable splitting techniques, one can deal with the two terms separately, and recent methods have investigated the usage of CNNs as prior. This is the case of [Rick Chang et al. 2017; Zhang et al. 2017] that show how a deep CNN trained for image denoising can effectively be used as prior in various image restoration tasks including SR. Instead of considering a denoising prior, Bigdeli *et al.* [2017] propose to use a prior based on an estimate of natural image distribution. These methods do not assume any knowledge on the degradation operation and rely on the prior to solve this ill-posed inverse problem. Although competitive, they do not outperform discriminatively trained SR neural networks.

The relative importance of image prior and reconstruction constraint was investigated by Efrat *et al.* [2013] who showed the importance of *blur kernel estimation* in the SR problem. A strategy already adopted by earlier works such as [Begin and Ferrie 2004] that used learning to recover camera point spread function (PSF) in SR problem. We can divide these approaches in two classes. In the first, the analysis is based on edges and contours in the image; For example, Qiao *et al.* [2006] propose an SVM to estimate the variance of a Gaussian blur kernel using features extracted with a Sobel operator, while Joshi *et al.* [2008] assume that contours in the image correspond to sharp edges that can be reconstructed, and the camera PSF is computed from these pairs of observed and predicted values. The second class of methods rely on image patch comparisons; Begin and Ferrie [2007] recover the camera point spread function by matching patches from the low resolution input to other patches from a training set of high resolution images; Michaeli and Irani [2013] take advantage of patch redundancy at different scales in the low resolution image to estimate the blur kernel the kernel. Interestingly, they also point the relation between the PSF and the blur kernel to use for SR.

Recently, some deep learning approaches have been proposed to tackle the more general case of *variable degradation kernel*. For instance, Zhang *et al.* [2018d] provide blur kernels as supplementary inputs to a super-resolution NN. The blur operation is modeled as an anisotropic Gaussian that is mapped to a new representation using a PCA. With the same objective of adapting the image synthesis to multiple degradations, Zhang *et al.* [2019] use a different degradation model where the blur operation is applied after a down-sampling, assumed to be bi-cubic. The super-resolution problem is solved by replacing the Gaussian denoiser prior with an image super-resolver prior. Both methods achieve good results but require the knowledge of the blur kernel and are thus unfitted for blind super-resolution.

In the blind setting, Shocher *et al.* [2018] train a network specifically for each image after recovering the blur kernel using patch repetition assumption [Michaeli and Irani 2013]. Concurrent to our work, Gu *et al.* [2019] propose to automatically estimate the kernel in the restricted case of an isotropic Gaussian blur. First, a neural network estimates the kernel variance directly from the low resolution image. Then, in an iterative process, another network computes the update step to apply on the kernel to reduce the artifacts. Their solution is different from our work in important ways; The space of kernels we consider is not limited to isotropic Gaussians. In such complex setting, the initial kernel estimation step [Gu et al. 2019] becomes more challenging. In addition to this, with a larger kernel space, predicting the update step is likely to lead to a local minimum. This shows the importance of our kernel discriminator that can evaluate high resolution output for any kernel.

3 METHOD

Our objective is to solve the blind super-resolution problem where given a low resolution image I_l, we would like to estimate the corresponding high resolution image I, such that:

$$I_l = (I * k)_{\downarrow s} \ , \tag{1}$$

where k is the unknown degradation kernel. The down-sampling operation $\downarrow_s$ depends on the considered scaling factor s.

In the non-blind setting, the kernel k is known and it is possible to have the synthesis process adapt to it [Zhang et al. 2018d]. We follow a similar strategy to build a super-resolution convolutional neural network (referred to as the *generator* in this paper) that takes into account the degradation kernel. This information allows the generator to be more flexible in the range of low resolution images it can handle (section 3.1).

The main challenge in blind super-resolution is to recover a high resolution image when the degradation kernel k is unknown. Our main contribution resides in the strategy we employ to recover this kernel. It is in particular based on the observation that providing the incorrect kernel in the synthesis process generates artifacts in the estimated high resolution image. Using another CNN (referred to as the *kernel discriminator*), we are able to detect these artifacts in the generator output and therefore identify whether the correct kernel was used. More details about this part can be found in Section 3.2.

With this discriminator, it becomes possible to recover the original degradation kernel by minimizing the errors detected in the generator output. This relies on the optimization process described in Section 3.3.

3.1 The Generator - Degradation Aware Super-Resolution

The degradation aware super-resolution approach we use consists of the kernel mapping network $\mathcal{F}_k$ and the generator $\mathcal{F}_g$ illustrated in figure 2. First, the degradation kernel k is mapped to a latent representation q_k. By considering the same degradation at each pixel location, we obtain the degradation map ρ. Next, the map ρ is passed to the generator along with the low resolution image I_l to produce a high resolution image.

In the *kernel mapping* step we propose to use a neural network $\mathcal{F}_k$ with parameters λ_k. Before providing the kernel k to the *generator*, we compute its low dimensional representation q_k:

$$q_k = \mathcal{F}_k(k \mid \lambda_k). \tag{2}$$

With this strategy, we have the possibility of learning a mapping more adapted to the super-resolution task than using a principal component analysis [Zhang et al. 2018d]. In practice, $\mathcal{F}_k$ is a two-layer dense network that takes as input the kernel k in vector form, obtained by row concatenation, and maps it to the reduced vector latent representation q_k.

The *Super-Resolution generator* predicts an estimate I^* of the high resolution image from the low resolution image I_l and a per pixel degradation map ρ of same size:

$$I^* = \mathcal{F}_g(I_l, \rho \mid \lambda_g). \tag{3}$$

In the case of a single degradation kernel k, its latent representation q_k is repeated for each pixel location. If different kernels k_i are used, per pixel or per region, we apply the kernel mapping transformation described above to each kernel separately and obtain the latent-space representations q_{k_i} that we assemble into the degradation maps ρ. Since we supply the degradation information as spatial feature maps, the kernel can vary in different parts of the image. This lets us handle the case of composited content, a very important part of real image-production pipelines.

The architecture of our generator is inspired by that of [Wang et al. 2018]. We use a sequence of dense compression units as the

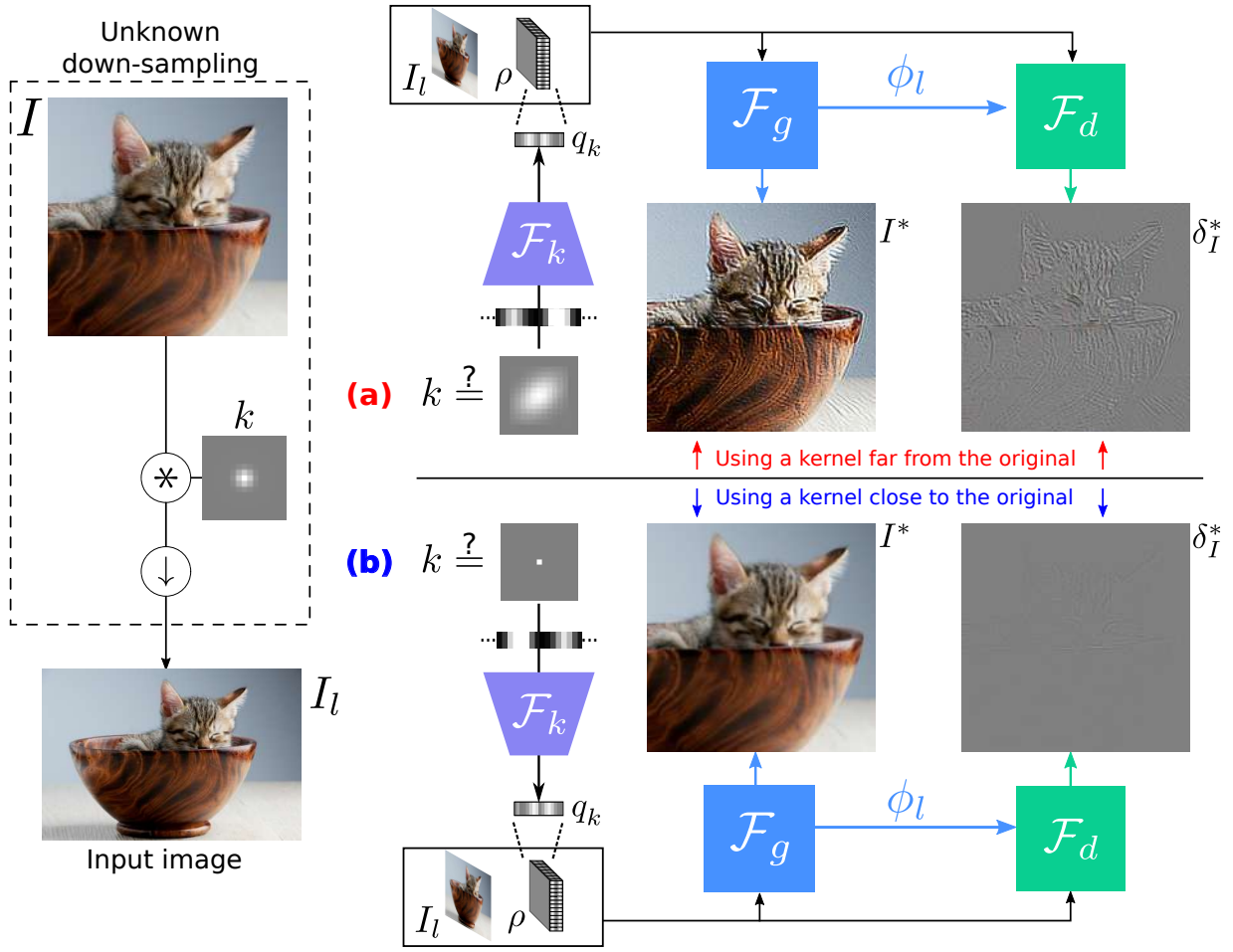

Fig. 2. **Overview.** In blind super-resolution, the degradation kernel k applied on the high resolution image to obtain the low resolution image I_l is unknown. Our pipeline is duplicated for two different kernels (a) and (b): the degradation-aware *generator* ($\mathcal{F}_g$) computes a high resolution output according to the provided blur kernel k. We note that a NN $\mathcal{F}_k$ is used to map the kernels to a low dimensional representation. The two kernels will result in different high resolution estimates. The kernel (a) farther from the unknown original degradation leads to more artifacts. To detect this, we propose a *kernel discriminator* network ($\mathcal{F}_d$) predicting the error due to using the incorrect kernel. By taking advantage of these two networks, we can express kernel estimation as finding the blur kernel resulting in the least amount of errors and artifacts in the predicted high resolution image (See text for details). Photo Credits: Pixabay/pexels.com.

core of the generator. The network predicts a residual image that is then added to a bicubicly upsampled image to produce the output I^*. Training the generator can be formally expressed as

$$\lambda_g^*, \lambda_k^* = \arg\min_{\lambda_g, \lambda_k} \mathbb{E}_{I\sim p_I, k\sim p_k} \left[\mathcal{L}\left(I, \mathcal{F}_g(I_l, \rho \mid \lambda_g)\right) \right]. \quad (4)$$

During training, we consider a single degradation k for the entire image which is randomly sampled among a set of realistic kernels. We approximate the distribution of real images p_I by random sampling in a data set of high resolution images. We used the ℓ_1 loss for training but other loss functions can be similarly considered.

3.2 The Kernel Discriminator

If the degradation kernel is known, the previously described super-resolution network can synthesize an estimate of the original high resolution image. This information is however not available in a blind setting and we observe that using the wrong kernel leads to noticeable artifacts in the synthesis. Figure 2 illustrates the results obtained using the generator with two different kernels. In the first case, a kernel far from to the original is used. The resulting high resolution image contains several artifacts. In the second case, the difference with the original kernel is smaller and the generator is able to recover a sharp image. In short, the generator results depend on the correctness of the provided degradation prior.

To take advantage of this, we propose using a second network, further referenced as the *kernel discriminator*, to estimate the errors in the generated image I^*. We note δ_I the pixel-wise residual on the synthesized high resolution image that should be predicted by the discriminator:

$$\delta_I = \mathcal{F}_g(I_l, \rho_{\text{GT}} \mid \lambda_g) - \mathcal{F}_g(I_l, \rho \mid \lambda_g), \quad (5)$$

Fig. 3. **Super-resolution with spatially varying degradation.** In this example we consider a gaussian blur kernel with a standard deviation increasing proportionally to the distance from the image center. Estimating a *single kernel* kernel for the entire image is not optimal, showing both over sharpening artifacts (near the image center) and blurring (near the borders). In this case a spatially varying estimation of the kernel is required to achieve best results. In addition to the output images, we provide a visualization of the estimated and ground truth degradation maps; Gray levels indicate standard deviation values. Photo Credits: Ella Olsson/pexels.com.

where ρ_{GT} is the ground-truth degradation map used to generate the low resolution image I_l, while ρ is a degradation map that we sample from our kernel distribution at training time or that we optimize at test time.

The architecture of the discriminator is similar to that of the generator. It takes as input the low resolution image I_l and the degradation map ρ. Instead of using the final output of the generator I^*, we provide the last feature map extracted by the generator (denoted ϕ_l):

$$\delta_I^* = \mathcal{F}_d(I_l, \phi_l, \rho \mid \lambda_d). \tag{6}$$

Using a fixed trained generator, we train the discriminator with the same dataset of high resolution images.

$$\lambda_d^* = \arg\min_{\lambda_d} \mathbb{E}_{I \sim p_I}\left[\mathcal{L}(\delta_I, \mathcal{F}_d(I_l, \phi_l, \rho \mid \lambda_d))\right]. \tag{7}$$

At test time, our goal is to find ρ such that δ_I^* is as close to zero as possible.

3.3 Optimizing for the Degradation Kernel

After defining the generator and the kernel discriminator in our pipeline, we now have all the required elements for kernel estimation. With the generator, we have an adaptable synthesis process that is expected to produce the best results when providing the correct degradation operation. The kernel discriminator on the other hand is trained to predict the errors that are present in the synthesis and hence discriminate between the degradation kernels. It will mostly identify regions with artifacts resulting from using the wrong kernel which typically appear around object contours and textured regions with high frequent details. This predicted residual can not be used directly to produce a corrected high resolution image as it will mostly smooth out the artifacts without producing a sharp image. Instead we will use the predicted error as an objective function that is minimized by finding the correct degradation kernel.

Formally, the kernel optimization can be written as

$$\rho^* = \arg\min_{\rho} \left\| \mathcal{F}_d(I_l, \phi_l, \rho \mid \lambda_d) \right\|_1, \tag{8}$$

where the locally adaptive kernel latent map ρ is estimated for the low resolution image I_l. The advantage of this formulation is to allow the estimation of a spatially varying degradation. In the simpler case of a single blur kernel k for the whole image, the optimization can be written with respect to a single latent representation q_k.

There are several options to practically solve this problem. Here we solve it in a two-stage approach. As the evaluation of equation 6 is fast, we first sample uniformly the kernel space and evaluate the error for each. The kernel with the lowest error is selected as starting value ρ, which is further optimized in the second stage of our procedure. In this stage, we optimize kernel latents through an iterative procedure where gradient descent is applied on the latents according to

$$\rho^* = \rho - \eta \nabla_\rho \mathcal{L}(I_l, \rho). \tag{9}$$

$\mathcal{L}(I_l, \rho)$ corresponds to the loss function defined by equation 8. This is similar to the strategy used for model training and the weights η to be applied on the gradients are obtained from the Adam optimizer [Kingma and Ba 2014]. The optimization can be done in several configurations. We can have a single degradation kernel for the whole image or have one kernel per pixel in the image. This lets us handle the case of spatially-variant degradations. We can also constrain the optimized kernel to remain in our kernel space (see section 4.1). For all the results in the paper, we perform the local optimization per image patch, as we found it to be more robust.

Figure 3 shows an image down sampled with a spatially varying degradation. We used a gaussian kernel with a standard deviation proportional to the distance from the image center. In this case, estimating a *single kernel* for the image leads to both blurring and over sharpening artifacts. A spatially adaptive estimation of the kernel is necessary to achieve good results. Although closely resembling the ground truth, the estimated degradation map has some

differences. This is expected as correctly modeling the blur kernel is only considered an intermediate step; The kernel estimation may be incorrect as long as it does not impact the final image quality.

4 PRACTICAL APPLICATIONS OF BLIND SR

The blind approach that we propose facilitates the usage of SR in practical scenarios. First, we describe how we adapt the parameterization of the kernel space to address typical down sampling operations (sec. 4.1). Then we show how we enable both local and global user fine-tuning (4.2) and how our method can be applied to composited content (sec. 4.3).

4.1 Kernel Space Representation

As expressed by Equation 1, the image degradation process is described as a blurring operation followed by a downsampling. The blur kernel used has a great influence on the final result and being able to handle a wide range of kernels translates to a much more general SR algorithm that works optimally in more cases. This blurring operation can be implicit, for example in the case of raw camera footage or in rendered content. But it can also happen as part of the visual effects pipeline. In this case, the blurring operation is often selected by an artist from a common set of filtering operations based on visual preference.

Our objective is to adapt to these different situations and in our tests, we select a set of *base kernels* related to common scaling operations available in most image and video processing software. Specifically these are: impulse, disk, bicubic and Lanczos. To further expand the capacity of the kernel space, we convolve these base kernels with a 2d anisotropic Gaussian. Figure 4 shows several samples from the considered kernel space.

As described in Section 3.1, we map the degradation kernel to a latent space representation using a fully-connected neural network (NN). We chose a neural network over the PCA favored by Zhang *et al.* [2018d]. As opposed to separately computing basis vectors that allow to minimize the kernel reconstruction error given a lower dimensional description, we jointly learn a specialized mapping to a compact representation that helps the upscaling task. As such, our mapping can extract more relevant information from the kernel. We show the difference between the two approaches in Figure 5 when considering two extreme kernels. On the first line, a very narrow impulse kernel is used to down sample the image whereas in the second line a much larger kernel was used. This second kernel is obtained by increasing the Gaussian standard deviations. Despite using the same generator architecture and training procedure, we can see a clear difference between the two options. The network using the PCA reduction performs worse in the extreme case of the extended kernel, while the NN mapping result remains sharp.

4.2 User Interaction

Once the optimization process has determined which kernel produced the best looking image, it is still possible for a user to modify its parameters to keep improving the visual quality of the result. This can be done since we define our kernels as convolutions of a base kernel with an anisotropic Gaussian kernels which itself is defined by three parameters: two standard deviations and one

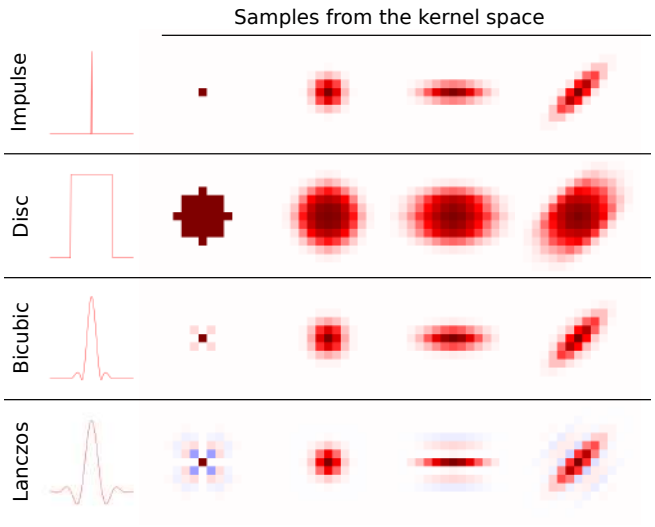

Fig. 4. **Kernel Space.** The kernels we use are convolutions of classic filters with anisotropic Gaussians of varying standard deviations and orientations.

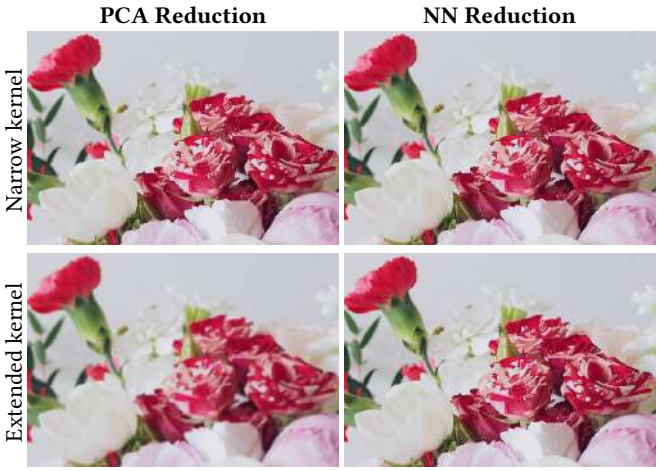

Fig. 5. **Comparison of PCA vs. NN kernel reduction.** PCA reduction performs well for simple kernels but fails when handling more complex degradations with more blurring. The neural network mapping performs much better in difficult settings. Photo Credits: Leonardo Wong/unsplash.com.

orientation angle. The Gaussian kernel gives us more control over the degradation and by acting on the standard deviations, a user can modify the high resolution result and easily fine-tune its sharpness both globally and locally.

We created a painting-like interface for refining our high resolution output locally. As shown in Figure 6, it is possible to "paint" the desired local contrast levels. The orange zone in the first column represents the brush stroke made by a user. We then increase the standard deviation of the kernels in that orange zone before feeding the degradation maps to the generator. This leads to an increased sharpness of the result in that area. This effect is very visible on the head of the parrot. It is also possible to decrease the standard

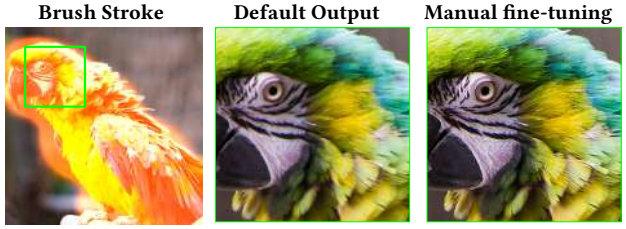

Fig. 6. **Examples of user-controlled refinement of SR result.** In the first column, we see the selected regions for fine-tuning (in orange). By locally increasing the standard deviation of the kernel provided to the generator, we cause the SR output to be locally much sharper. Outside the brush stroke, the image stays unchanged. Photo Credits: Andrew Li/unsplash.com.

deviation and make the image locally less sharp. Outside the brush stroke, the image stays the same. The generator does not diffuse local changes to other areas in the image.

4.3 Spatial Composition

Representing kernel information with spatial feature maps permits a lot of freedom in adapting the image degradation locally. To illustrate the advantage of this spatial adaptation, we compare our results with the typical upscaling approaches used in a production pipeline such as *Nuke's TVIScale*. This is a total variation inpainting based approach for upscaling. We also compute the results obtained using an SR generator that does not have any information about the kernel (referenced as *No-Kernel Generator*).

Figure 7 presents two compositing tests. The first sequence is based on the sample project from the open source compositing software *Natron* and the second uses images from the open-source Blender movie *Tears of Steel* Our approach is able to locally adapt the kernels and hence targets the case where the composition mask is unknown. If the mask is provided, it can be used to optimize a kernel for each region before combining them to produce the final output. Comparisons with the different approaches are provided for the zoomed in regions. The kernels estimated by our approach using the masks are provided in the rightmost column.

We did not down sample the image in the case of the spaceship and used the original frames. This is a concrete example of an image source with unknown properties where our algorithm is able to recover the kernel properties and produce good upscaling results both on the foreground and the background of the image. In the second scene, the main character is composited on a rendered background. Each part of the image is down sampled independently according to the mask. Here as well, our method produces the sharpest result. It also manages to recover kernels very close to the ones we used. These results show the benefits of using our locally adaptive SR algorithm that recovers more details than classic upscaling tools or the *No-Kernel* generator.

5 COMPARISONS AND DETAILED EVALUATION

We present a detailed evaluation of our approach and comparisons with state of the art methods in blind SR. Both the generator and the kernel discriminator have the same architecture based on the super-resolution network proposed by [Wang et al. 2018] (see supplemental material for details).

We train 3 type of generators. First a generic degradation aware generator is trained for all the degradation kernels described in Section 4.1. Second, a *No-Kernel* generator is trained on the same set of degradations but without any information regarding the kernels. This generator is our baseline to evaluate the importance of having a degradation aware network. Finally, specialized degradation aware generators are trained for each *basic* kernel category. For example, the generator specialized in bicubic kernels is trained on a bicubic kernel convolved with random anisotropic gaussians. After this, we train 2 types of kernel discriminators, corresponding to the generic and specialized degradation aware generators. The discriminators are trained using the same procedure as their corresponding generators. The generator weights are kept constant while training the discriminator. All our models are trained for 2× upscaling in 10 days using the DIV2K [Timofte et al. 2017] dataset, which contains 800 high resolution images. During training, a blur kernel is randomly sampled in the considered kernel space to obtain the low resolution image. For the quantitative evaluation, we used the *BSD100* dataset [Arbelaez et al. 2010] and the *Set14* [Zeyde et al. 2010]. All quantitative evaluations using PSNR and SSIM as error measure are conducted on the luminance channel as commonly done in existing literature. In addition to this we use the perceptual error metric (LPIPS) proposed by Zhang *et al.* [2018b].

Processing a Full-HD image with our x2 upscaling framework and local patch optimization, on an NVIDIA GTX 1080Ti GPU, takes around 30 seconds for the kernel grid search initialization and 2 minutes for the optimization process (50 iterations using Adam optimizer with a learning rate of 0.1). We found the initial grid search to be important to avoid local minima and maintain low runtime. The actual SR generation process can be done at an almost interactive rate. If we want to upscale a set of images from the same source, we could estimate a kernel for one image and reuse it for the others.

5.1 Qualitative evaluation

To showcase the benefits of our method, we selected a set of high resolution images online and downsample them using classic kernels. The obtained low resolution images are then upscaled using different approaches (Fig. 8). Our algorithm achieves the best result in all the considered cases. For example, the *rocket* illustration in the first row was down sampled using an impulse kernel. All methods generate more or less aliasing on the contours whereas ours is able to avoid this while producing a sharper image. On the *Taxi* image, a disk kernel was used and thus the details are blurred in a more significant manner. We can see that our solution nicely recovers the details of the numbers on the car speed dial. Using a bicubic downscaling is the most advantageous setting for the other methods but we can still see improvements as we are able to better recover the freckles on the skin contrary to the No-Kernel generator that over-smooths the details and the TVIScale that produces a noisier image. On the last example, *Temple*, we are able to better reconstruct the structure of the arcades and produce less aliasing in general.

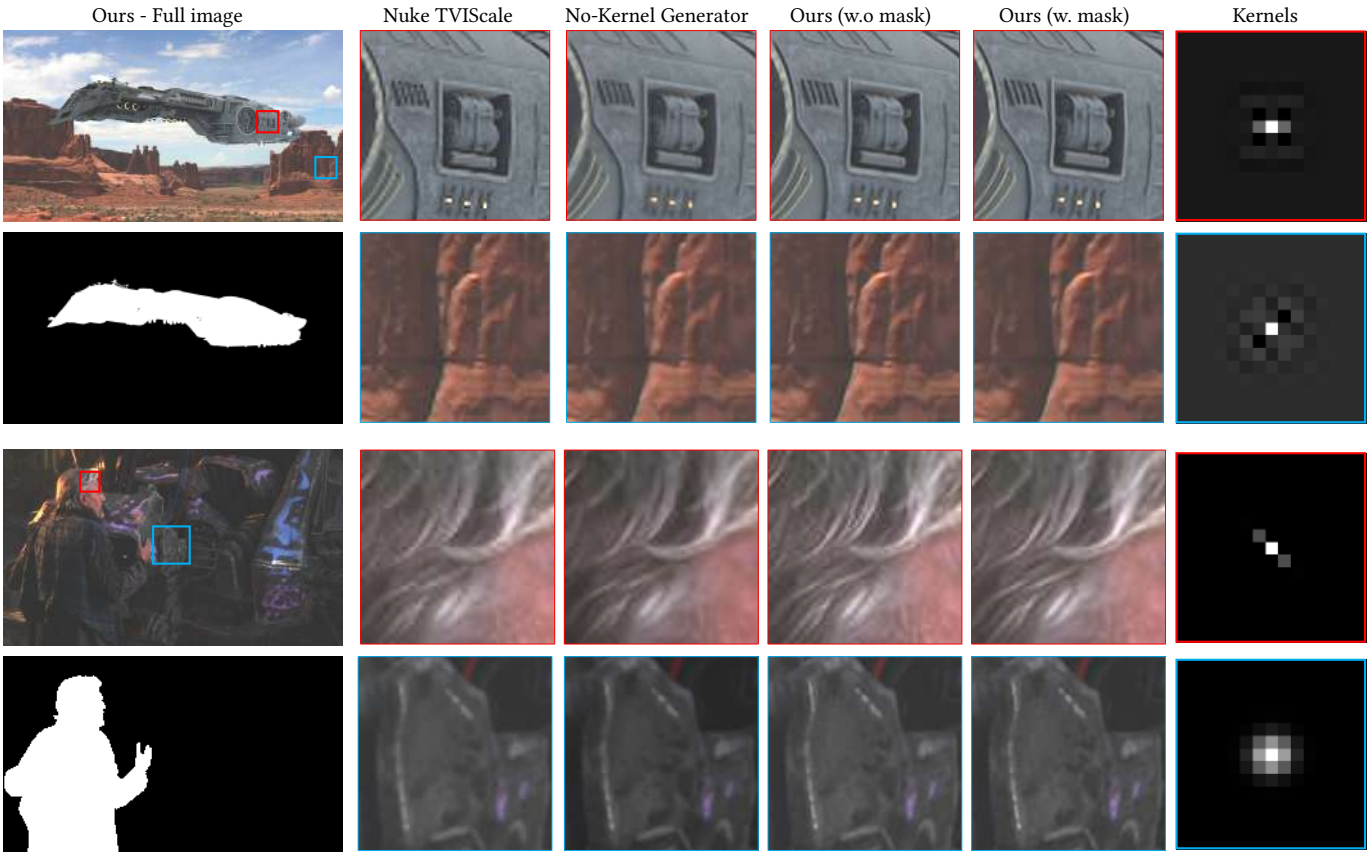

Fig. 7. **SR on composited content.** We can see the interest of using our locally adaptive SR algorithm on these sequences combining real footage with rendered content. In the case of the spaceship we used the original frames where the down sampling is unknown. In the second sequence, each image region is down sampled independently according to the mask. Our approach is able to locally adapt kernel estimation and achieve better results than both TVIScale and No-Kernel Generator. When the compositing mask is available, we estimate a single kernel for each region which results in slightly sharper images (more visible on the spaceship). We provide the kernel estimated for each region using the mask in rightmost column. Photo Credits: Spaceship by Francois Grassard (CC-BY) and *Tears of Steel* by (CC) Blender Foundation | mango.blender.org.

In Figure 9, we show 4× upscales of images taken with a DSLR camera and a mobile phone. The input images have not been downscaled, so the degradation is unknown and derives from the cameras' optics and imaging pipelines. Our upscaled results are sharper and have fewer artifacts than those of a state-of-the-art SR method trained to assume bicubic downsampling.

5.2 Comparisons with blind SR methods

We compare our approach with existing blind SR methods and use the same test set as ZSSR [Shocher et al. 2018]. The authors have graciously provided the low resolution images obtained by downscaling the HR images using random Gaussian kernels. Please refer to the original paper for details regarding the kernel generation. In addition to ZSSR, the comparison also includes two other methods that are state-of-the-art: BlindSR [Michaeli and Irani 2013] and EDSR [Lim et al. 2017].

We present representative results in the visual comparison of Figure 10. EDSR is trained for bicubic down sampling and thus can not adapt to new degradation operations. ZSSR combines the advantages of deep neural networks with the kernel estimation from BlindSR. ZSSR improves over previous methods but requires training a SR network for each image using the estimated kernel. Our solution produces better results thanks to a more precise blur kernel estimation and a more powerful degradation aware generator. On the first row our, results are sharper while in the second we also see that the produced high resolution image does not contain aliasing artifacts contrary to ZSSR.

The quantitative evaluation in Figure 11 is using PSNR and SSIM as error measure and shows the superiority of our solution by a clear margin. In the blind setting we obtain more than 1db improvement over the best performing approach ZSSR. We are even able to outperform ZSSR results based on the ground truth kernel.

5.3 Detailed evaluation

We consider several standard filtering kernels — impulse, cubic, Lanczos and disk — convolved with an anisotropic 2d Gaussian as

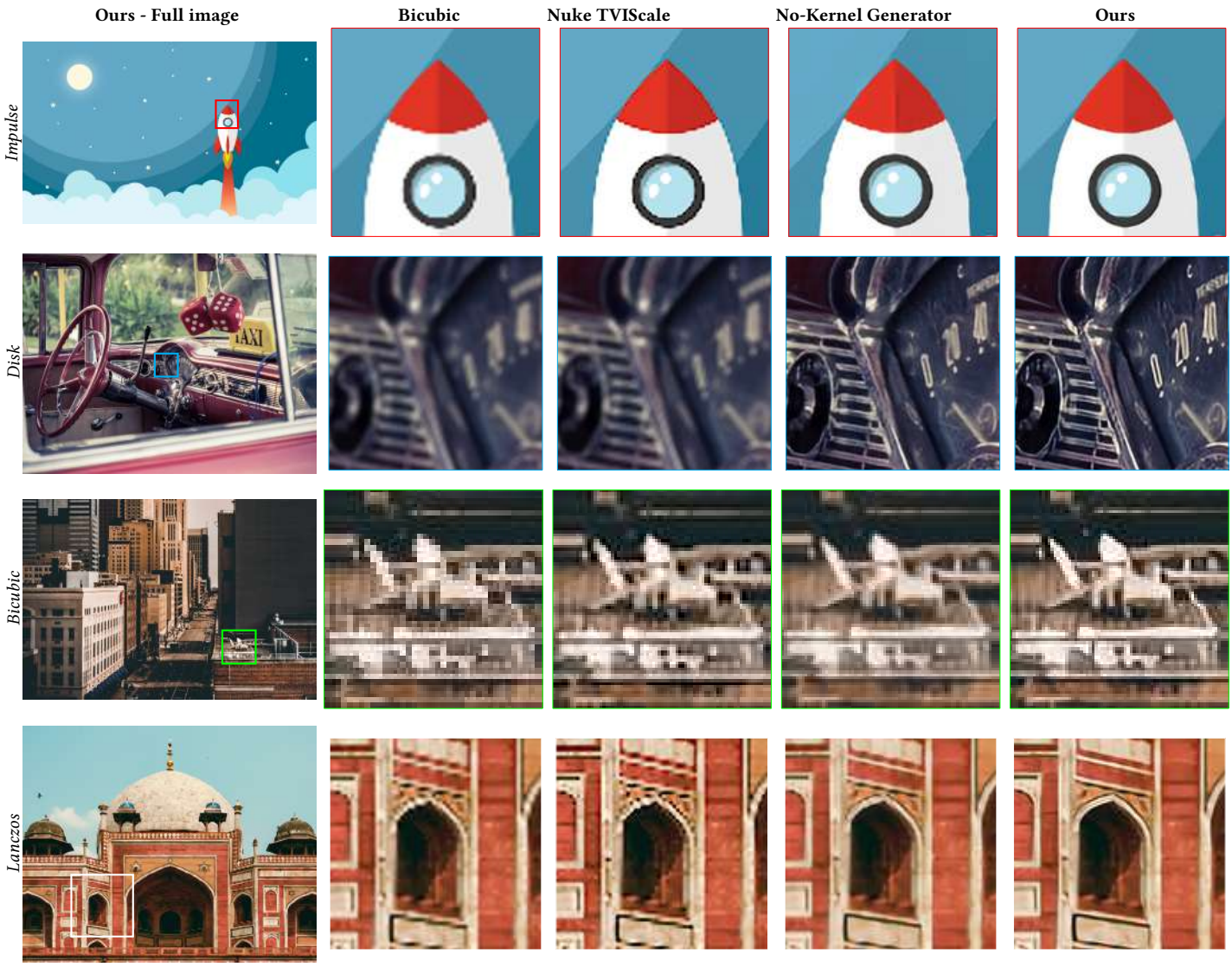

Fig. 8. **Results on classic down-sampling kernels.** For each row, the leftmost column indicates the kernel that was used to create the low-resolution image. We include results from different approaches for visual comparison. This includes the most commonly used upscaling tool node in Nuke (*Nuke TVIScale*). The *NoKernel generator* is a neural network trained for all down-sampling operations but without the knowledge of the kernel. Our approach automatically estimates the kernel and outputs the best upscaling results on a large variety of content. Photo Credits (from top to bottom): Yuri_B/pixabay.com; Pixabay/pexels.com; Benjamin Suter/unsplash.com; Dewang Gupta/unsplash.com.

illustrated in figure 4. In this detailed evaluation, our objective is to understand the differences between a specialized neural network and a more general one for both the blind and the non-blind settings.

We use the *Set14* images and for each *base kernel*, sample several parameters for the Gaussian. After down sampling, the images are upscaled using the different generators in both blind and non-blind settings. For reference we also provide results for the *No-Kernel* generator. The evaluation is presented in Figure 12 and uses PSNR and the *Learned Perceptual Image Patch Similarity* (LPIPS) from [Zhang et al. 2018b] as error measures. A higher PSNR is better while a lower LPIPS is better.

We can extract several important pieces of information from these results: First, comparing the results of the *No-Kernel* network with the other columns, we can see that not using any kernel information is detrimental in all cases. Information about the degradation helps in every case (blind and non-blind) and both PSNR and LPIPS values show clear improvements. Second, our results when operating with the kernel discriminator in a completely blind setting are close to those obtained with knowledge of the ground-truth kernel. Finally, the generators specialized in one type of degradation perform only slightly better than the generic network. We can note more difference in the blind case for the most challenging kernel (*Disk*).

Fig. 9. **SR results (4×) for non downscaled images.** Our solution is able to achieve better results by reducing the artifacts present in the high resolution images. Images in the first two rows are captured using a DSLR camera whereas the last rows correspond to mobile phone images.

In Figure 13, we consider the setting of explicit bicubic downsampling on the input and compare our approach to methods specifically trained for this case on the BSD100 data set. Since our focus was to explore the blind setting, we have opted for using a significantly condensed version of the ProSR architecture [Wang et al. 2018]. As a result, there is a gap in PSNR compared to the original version but also one order of magnitude less parameters. When comparing our architecture once specifically trained for bicubic downsampling without injecting degradation maps and once trained for the blind case, we notice that both achieve a similar quality. This indicates that the way we are making degradation information available to the network does not have an impact on reconstruction quality. Note that the number of parameters in the blind case is only higher because we also count the number of parameters in the kernel discriminator which is only used to estimate the kernel.

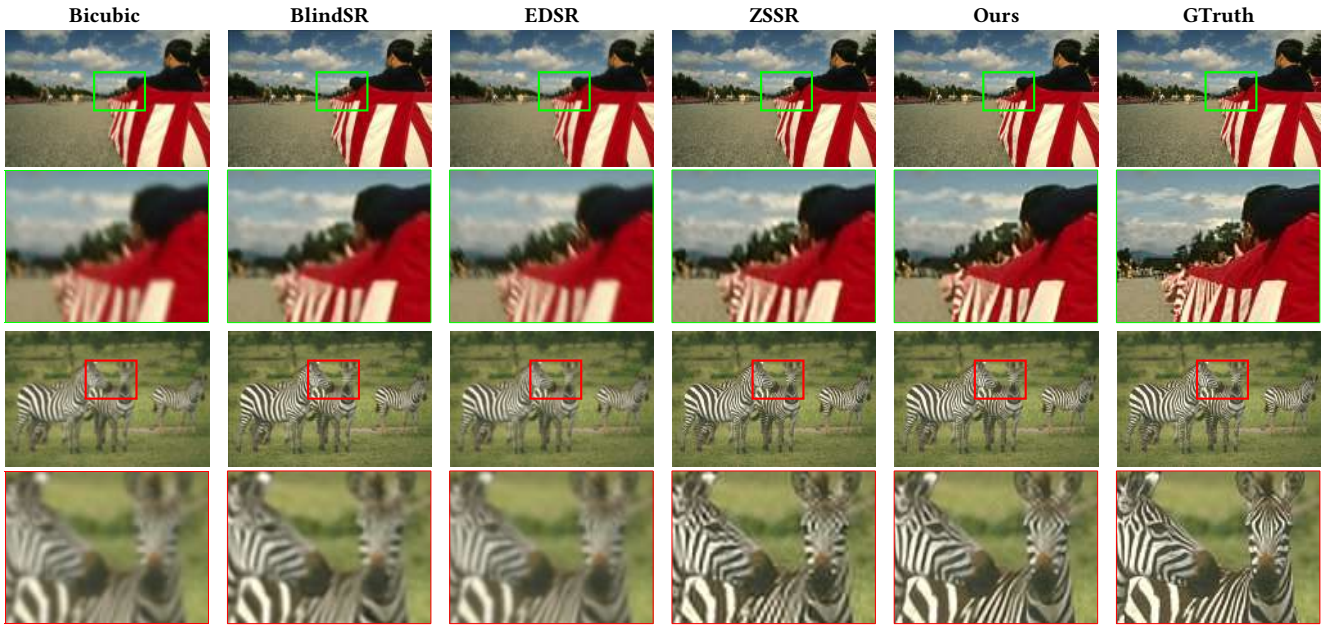

Fig. 10. **Visual comparison with existing SR Methods.** These images are taken from the dataset provided by [Shocher et al. 2018]. Our algorithm produces the sharpest results of all the methods presented here. On the zebra image, we can also see that we restore strong edges correctly and avoid the aliasing present in ZSSR result. Input images: Berkeley Segmentation Dataset.

Method	PSNR	SSIM
VDSR [Kim et al. 2016]	27.72	0.764
EDSR [Lim et al. 2017]	27.78	0.766
BlindSR [Michaeli and Irani 2013]	28.42	0.783
ZSSR (blind) [Shocher et al. 2018]	28.81	0.831
ZSSR (w. kernel)	29.68	0.841
Ours (blind)	**29.92**	**0.846**

Fig. 11. **Quantitative evaluation.** Our method consistently outperforms other state-of-the-art algorithms on the BSD100 dataset downsampled with random kernels introduced by [Shocher et al. 2018].

5.4 Limitations and Failure cases

The main limitation of the proposed approach is related to the considered kernel space. The classical kernels we chose as basis are kernels commonly used for image processing tasks, but it is possible that the real kernel is far from this space. To investigate this, we include two experiments with degradations *not seen* during training; In the first, we use the *Welch* kernel which corresponds to a degradation relatively close to our basis. In the second, we use a kernel corresponding to motion blur, significantly different from any degradation seen during training.

Figure 14 shows the results obtained when using the *Welch* kernel. Although the generator was not trained on this particular degradation, the results are better than the *no-kernel* alternative and with sharp details better restored. Figure 15 illustrates a much more challenging scenario corresponding to motion blur. In this case, even using the ground truth kernel for the generator leads to strong artifacts. It is interesting to note that, as our discriminator goal is to reduce artifacts in the image, even in this case the selected kernel does not create artifacts and leads to a visually more pleasing image.

6 CONCLUSIONS

In this paper, we described a framework that is able to perform blind SR in a completely automated way. One key aspect is the kernel discriminator network that is able to analyze artifacts created by a degradation-aware SR network. In addition to this, the proposed optimization is able to estimate degradation both locally and globally. This is beneficial from a practical point of view as we can address upscaling composited content even in the case where the masks are unavailable. Thanks to our parametrization of the kernel space, we achieve even more flexibility by allowing local manual tuning of the sharpness of the results.

Both qualitative and quantitative comparisons show the superiority of the proposed solution over state-of-the-art methods in several scenarios. The detailed evaluation showed that providing information about the degradation only through the training data is not sufficient to train a neural network that can adapt well. In contrast to this, incorporating information about the kernel in the model allows for good adaptation to all degradations observed in the training data and works even reasonably well for unseen ones. This ability to generalize and to automatically detect degradations is an important step towards leveraging the full potential of deep learning based upscaling in more practical scenarios. To push these efforts even further in the future, other very relevant directions for research include enabling arbitrary non integer scaling factors and optimizing the network for efficiency gains.

Degradation type	(a) No-Kernel		(b) Type-specific (w. Kernel)		(c) Generic (w. Kernel)		(d) Type-specific (blind)		(e) Generic (blind)	
	PSNR	LPIPS	PSNR	LPIPS	PSNR	LPIPS	PSNR	LPIPS	PSNR	LPIPS
Impulse	31.42	0.145	32.77	0.105	32.80	0.104	31.88	0.118	32.00	0.122
Cubic	31.36	0.151	33.10	0.103	32.92	0.102	32.28	0.113	32.23	0.120
Lanczos	30.99	0.150	32.52	0.106	32.40	0.110	31.70	0.118	31.57	0.129
Disk	30.91	0.166	32.81	0.095	32.32	0.112	32.01	0.108	31.63	0.144

Fig. 12. **Detailed SR evaluation.** We performed a quantitative evaluation of different configurations with different types of kernels. All experiments were done on the *Set14* dataset. (a) Generator network with no knowledge of any degradation information. (b) Network specialized in a specific type of kernel with knowledge of the ground-truth kernel. (c) Generic network with knowledge of the ground-truth kernel. (d) Kernel discriminator specialized in the specific type of kernel in a blind setting where the ground-truth kernel is not given. (e) Generic kernel discriminator in a blind setting where ground-truth kernel is not given.

Method	PSNR	SSIM	Parameters
RCAN [Zhang et al. 2018c]	32.46	0.903	~ 14M
ProSR [Wang et al. 2018]	32.34	0.902	~ 10M
EDSR [Lim et al. 2017]	32.32	0.901	~ 43M
VDSR [Kim et al. 2016]	31.90	0.896	~ 5M
Ours (bicubic)	31.35	0.891	< 1M
Ours (blind)	31.27	0.890	< 2M

Fig. 13. **Comparison to non-blind super resolution methods.** We have used a significantly condensed version of the ProSR architecture. Therefore, our PSNR is lower even if trained specifically for bicubic downsampling. However, our blind approach achieves a similar quality as our generator trained for bicubic downsampling.

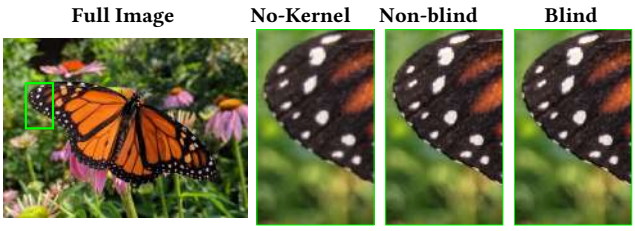

Fig. 14. **Unseen kernel close to our basis.** Although the Welch downsampling kernel was not used during training, the SR algorithm is able to adapt to it and produce good results both in non-blind and blind setting, outperforming the No-Kernel generator. Photo Credits: Jeebz K/unsplash.com

Fig. 15. **Unseen kernel far from our basis.** In this case a motion blur kernel is used. The generator was not trained with this type of kernels and cannot revert the degradation. The resulting upscaling has strong artifacts. Despite the generator limitations, our discriminator finds a conservative kernel that leads to an artifact free image. Photo Credits: Pixabay/pexels.com.

REFERENCES

Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. 2010. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence* 33, 5 (2010), 898–916.

Isabelle Begin and FR Ferrie. 2004. Blind super-resolution using a learning-based approach. In *ICPR*.

Isabelle Begin and Frank P Ferrie. 2007. PSF recovery from examples for blind super-resolution. In *ICIP*.

Siavash Arjomand Bigdeli, Matthias Zwicker, Paolo Favaro, and Meiguang Jin. 2017. Deep mean-shift priors for image restoration. In *Advances in Neural Information Processing Systems*.

Jose Caballero, Christian Ledig, Andrew Aitken, Alejandro Acosta, Johannes Totz, Zehan Wang, and Wenzhe Shi. 2017. Real-time video super-resolution with spatio-temporal networks and motion compensation. In *CVPR*.

Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. 2014. Learning a deep convolutional network for image super-resolution. In *ECCV*.

Chao Dong, Chen Change Loy, and Xiaoou Tang. 2016. Accelerating the super-resolution convolutional neural network. In *ECCV*.

Netalee Efrat, Daniel Glasner, Alexander Apartsin, Boaz Nadler, and Anat Levin. 2013. Accurate blur models vs. image priors in single image super-resolution. In *ICCV*.

Jinjin Gu, Hannan Lu, Wangmeng Zuo, and Chao Dong. 2019. Blind super-resolution with iterative kernel correction. In *CVPR*.

Neel Joshi, Richard Szeliski, and David J Kriegman. 2008. PSF estimation using sharp edge prediction. In *CVPR*.

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

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*.

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

Tomer Michaeli and Michal Irani. 2013. Nonparametric blind super-resolution. In *ICCV*.

Kamal Nasrollahi and Thomas B Moeslund. 2014. Super-resolution: a comprehensive survey. *Machine vision and applications* (2014).

Jianping Qiao, Ju Liu, and Caihua Zhao. 2006. A novel SVM-based blind super-resolution algorithm. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings*.

JH Rick Chang, Chun-Liang Li, Barnabas Poczos, BVK Vijaya Kumar, and Aswin C Sankaranarayanan. 2017. One Network to Solve Them All–Solving Linear Inverse Problems Using Deep Projection Models. In *ICCV*.

Mehdi SM Sajjadi, Raviteja Vemulapalli, and Matthew Brown. 2018. Frame-recurrent video super-resolution. In *CVPR*.

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

Assaf Shocher, Nadav Cohen, and Michal Irani. 2018. âĂIJZero-ShotâĂİ Super-Resolution using Deep Internal Learning. In *CVPR*.

Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang. 2017. Ntire 2017 challenge on single image super-resolution: Methods and results. In *CVPR Workshops*.

Yifan Wang, Federico Perazzi, Brian McWilliams, Alexander Sorkine-Hornung, Olga Sorkine-Hornung, and Christopher Schroers. 2018. A Fully Progressive Approach to Single-Image Super-Resolution. (2018).

Chih-Yuan Yang, Chao Ma, and Ming-Hsuan Yang. 2014. Single-image super-resolution: A benchmark. In *ECCV*.

Roman Zeyde, Michael Elad, and Matan Protter. 2010. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*. Springer, 711–730.

Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. 2017. Learning deep CNN denoiser prior for image restoration. In *CVPR*.

Kai Zhang, Wangmeng Zuo, and Lei Zhang. 2018d. Learning a single convolutional super-resolution network for multiple degradations. In *CVPR*.

Kai Zhang, Wangmeng Zuo, and Lei Zhang. 2019. Deep Plug-and-Play Super-Resolution for Arbitrary Blur Kernels. *CVPR* (2019).

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018b. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*.

Xinyi Zhang, Hang Dong, Zhe Hu, Wei-Sheng Lai, Fei Wang, and Ming-Hsuan Yang. 2018a. Gated Fusion Network for Joint Image Deblurring and Super-Resolution. In *BMVC*.

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