Pixel-Level Kernel Estimation for Blind Super-Resolution

JAIHYUN LEW1, EUIYEON KIM1, AND JAE-PIL HEO1,2, (Member, IEEE)

1Department of Artificial Intelligence, Sungkyunkwan University, Suwon 16419, South Korea
2Department of Computer Science and Engineering, Sungkyunkwan University, Suwon 16419, South Korea

Corresponding author: Jae-Pil Heo (jaepilheo@skku.edu)

This work was supported in part by the Ministry of Culture, Sports and Tourism and the Korea Creative Content Agency under Project R2020070002, and in part by the Institute of Information & Communications Technology Planning & Evaluation (IITP) funded by the Korean Government Ministry of Science and ICT (MSIT) under Project 2020-0-00973 and Project 2019-0-00421.

ABSTRACT Throughout the past several years, deep learning-based models have achieved success in super-resolution (SR). The majority of these works assume that low-resolution (LR) images are ‘uniformly’ degraded from their corresponding high-resolution (HR) images using predefined blur kernels — all regions of an image undergoing an identical degradation process. Furthermore, based on this assumption, there have been attempts to estimate the blur kernel of a given LR image, since correct kernel priors are known to be helpful in super-resolution. Although it has been known that blur kernels of real images are non-uniform (spatially varying), current kernel estimation algorithms are mostly done at image-level, estimating one kernel per image. These algorithms inevitably become sub-optimal in handling scenarios where an image is degraded non-uniformly. A divide-and-conquer form of approach, dividing an image into several patches for individual kernel estimation and SR can be a simple solution for this matter. Nevertheless, this approach fails in practice. In this paper, we address this issue by pixel-level kernel estimation. The three main components for training a SR framework based on pixel-level kernel estimation are as follows: Kernel Collage — a method for synthesizing non-uniformly degraded LR images, designed considering the coherency of kernels at neighboring regions while abruptly changing at times, the indirect loss — a novel loss for training the kernel estimator, based on the reconstruction loss, and an additional optimization — a scheme to robustify the SR network to minor errors in kernel estimations. Extensive experiments show the superiority of pixel-level kernel estimation in blind SR, surpassing state-of-the-art methods in terms of quantitative and qualitative results.

INDEX TERMS Blind super-resolution, blur kernel estimation, pixel-level estimation, spatially varying degradation.

I. INTRODUCTION Single image super-resolution (SISR) aims to recover a high-resolution (HR) image from a given low-resolution (LR) image. Deep learning-based SISR methods have accomplished remarkable results recently and a large portion of these methods are trained by using LR-HR image pairs. Since it is extremely expensive to obtain real LR-HR pairs, a majority of SISR methods use synthesized LR-HR pairs for training. LR images are typically assumed to be degraded from its HR version using predefined blur kernels, and LR-HR pairs are synthesized based on this assumption.

It has been shown that the blur kernels vary for each image [1], [2] and that an accurate blur kernel prior can greatly improve performance SISR [3]. Accordingly, there have been several attempts to estimate the blur kernels [1], [2], [4]. Current kernel estimation methods mostly estimate one kernel per image, based on an assumption that LR images are degraded uniformly — same kernel applied to all regions within an image. However, blur kernels of real images are not always uniform [5]–[7].

Although the problem of kernel estimation on non-uniformly degraded images has been discussed in the field of deblurring [8]–[10], it has not received sufficient attention in SR. In the field of deblurring, kernel estimations on non-uniform degradation are mostly done in a divide-and-conquer manner — dividing a whole image into several...
patches and applying kernel estimation algorithms to those patches individually. This patch-wise strategy can be utilized in SR, but this has a contradictory limitation that the kernels can still vary across pixels within a patch.

In this paper, we focus on pixel-level kernel estimation. To enable pixel-level kernel estimation, we introduce a non-uniform degradation method, Kernel Collage. In Kernel collage, diverse kernels are used within a single image in a spatially coherent manner, while being inconsistent at times. Since pixel-level kernel estimation cannot obtain information from other regions of the image, we suggest a novel loss to learn kernel estimation via reconstruction error between super-resolved images using the ground-truth (GT) kernels and predicted kernels. Once the SR network and the Predictor are fully trained, the SR network is re-trained with the pre-trained Predictor’s weights fixed to get robust to incorrect estimations on kernels.

Our main contributions are summarized as follows:

- We suggest Kernel Collage, a simple yet effective degradation method to train a SR framework based on pixel-level kernel estimation.
- We introduce a novel indirect loss for pixel-level kernel estimation, to relieve the issues of the conventional loss in image-level kernel estimation.
- Extensive evaluations show that employing pixel-level kernel estimation surpasses state-of-the-art techniques in blind SR, both on synthetic and real images.

II. RELATED WORK

A. SUPER-RESOLUTION

1) CNN-BASED SUPER-RESOLUTION

Ever since Dong et al. [11] introduced the first Convolutional Neural Network (CNN) based super-resolution model, numerous CNN based models for super-resolution have been proposed [12]–[15]. Conventionally, the networks are trained with the pixel-wise $L_1$ or $L_2$ loss function using synthesized LR-HR image pairs. Furthermore, Spatial Feature Transform (SFT) layer was introduced to incorporate categorical conditions into account [16] and an internal learning-based approach has shown to be effective in recovering recurrent contents throughout an image [17].

2) BLIND SUPER-RESOLUTION

Blind SR aims to super-resolve an image where the degradation parameters, such as blur kernels are unknown. As it has been known that an accurate kernel prior can be very helpful in blind SR [3], most blind SR methods are done in a sequential combination of kernel estimation and super-resolution.

Michaeli and Irani [1] estimate the blur kernel based on the recurrence of small patches within a single image and take advantage of it in the SR process. The inevitable problem of kernel mismatch has been addressed by Gu et al. [4] and they try to solve the problem by iteratively correcting the predictions on kernels. Recently, KernelGAN [2] has been proposed to estimate the degradation kernel of the given image through internal learning. All of these works estimate a kernel at image-level, one kernel per image. They are all based on an assumption that the degradation process is uniform across an entire image [1], [2], [4], which is not always the case in practical scenarios [5]–[7].

B. NON-UNIFORM DEGRADATION

Non-uniform degradation refers to a method of which an image is degraded in a spatially variant manner throughout a single image. It has been told by prior arts [5]–[7] that the blur kernels are non-uniform in real scenarios.

In the field of deblurring, attempts on non-uniform kernel estimation have been studied [8]–[10]. These approaches try to solve non-uniform blur kernel estimation by dividing the given image into patches and conducting kernel estimations individually. In contrast, non-uniform degradation and kernel estimation has not been discussed sufficiently in super-resolution. SRMD [18] and UDVD [19] proposed SR networks that could handle spatially varying degradation, but these are for non-blind SR, of which the degradation parameters must be given.

To our knowledge, the only work considering non-uniform degradation in blind SR is [20]. This work uses a kernel discriminator to make corrections on kernel estimation and can handle spatially variant degradation. However, it differs from our work in several aspects. For one, the entire SR framework of [20] is trained with uniformly degraded LR images only. Hence, it never encounters a non-uniformly degraded LR image in the training phase and becomes sub-optimal to super-resolve a non-uniformly degraded LR image. Moreover, although their method can make pixel-level estimations on kernels in theory, they conduct kernel estimation and super-resolution at patch-level in practice and do not address the necessity of pixel-level kernel estimation.

Current kernel estimation-based image restoration techniques for non-uniform degradation are mostly done at patch-level and attempts to aggregate the results afterward. In this paper, we propose a method to train a SR framework based on pixel-level kernel estimation that can handle non-uniformly degraded LR images without dividing the input image into multiple patches.

III. MOTIVATION

Current state-of-the-art kernel estimation methods typically estimate one kernel from an image or conduct kernel estimation in a patch-wise manner. In this section, we aim to demonstrate the weakness of image-level kernel estimation and emphasize the necessity of pixel-level estimation.

We have done a simple experiment using IKC [4] and DAN [21] to support our argument. Both algorithms estimate one kernel for an image and super-resolve the image based on it. As Figure 1 shows, IKC and DAN shows different SR quality at different regions — it still has artifacts and over-sharpens certain areas, whereas some regions are still...
left blurry. Since blurry or over-sharpening artifacts appear when the variance of the estimated kernels is smaller or greater than the actual kernels respectively [4], this example strongly indicates that the blur kernels are different region-by-region in real images and that estimating one kernel for an image is sub-optimal in practice. Hence, we aim to train a SR framework based on pixel-level kernel estimation to handle the non-uniformity of blur kernels.

One might question if the non-uniformity of kernels cannot be just handled in a patch-wise manner, like the approaches from deblurring. In fact, kernel estimation-based SR algorithms can be utilized in that manner. However, there are limits to this form of approach. Most importantly, making kernel estimations at patch-level is problematic in that it assumes each patch to have been degraded uniformly. It becomes sub-optimal in situations where the patches are also degraded non-uniformly, which is highly the case in real images. Continuously dividing patches into smaller patches may be an option, but since image contents are essential to make estimations on kernels, the patches cannot be divided from a certain point in order to estimate kernels. Therefore, applying kernel estimation algorithms patch-wise cannot fully manage the non-uniformity of kernels, and estimations on kernels should be made at pixel-level, one kernel per pixel, directly from a single image.

IV. PROPOSED METHOD

We propose a method to train a pixel-level kernel estimation model capable of handling non-uniformly degraded LR images. We discuss the strategy of non-uniform degradation to utilize in the training phase and present our approach for pixel-level kernel estimation and SR.

A. NON-UNIFORM DEGRADATION

In SISR, it has been dominantly assumed [2], [4], [18], [22], [23] that a LR image ($I^{LR}$) is degraded from a HR image ($I^{HR}$) by a degradation process of blurring, sub-sampling ($\downarrow$) in a uniform manner, followed by an additive noise term $n$:

$$I^{LR} = (I^{HR} \otimes k) \downarrow_s + n$$  \hspace{1cm} (1)

where $k$ stands for the blur kernel, $s$ for the scale factor and $\otimes$ for convolution operation. In this work, we disregard the noise term (i.e. $n = 0$), as many prior works did.

On the other hand, to expose a non-uniformly degraded LR image to our framework, we use a kernel map $K$ instead of a single kernel $k$ in the degradation process. Kernel map is a map that indicates which kernels to apply on the HR image when generating each pixel of the LR image. The non-uniform degradation process can be expressed in the following formula:

$$I^{LR}_{p,q} = (I^{HR}_{p,q} \otimes K_{p,q}) \downarrow_s$$  \hspace{1cm} (2)

where $p$ and $q$ respectively denote each pixel’s vertical and horizontal coordinates of the LR image and $\otimes$ denotes dot product. Each pixel of the LR image ($I^{LR}_{p,q}$) is degraded with a kernel of size $h \times h$ from the same coordinates of the kernel map ($K_{p,q}$). $I^{HR}_{p,q}$ represents a patch of size $h \times h$ from the HR image where the center coordinates are $(p \times s, q \times s)$.

1) KERNEL MAP GENERATION

The main point of non-uniform degradation is to degrade various regions of an image with differing kernels and according to the non-uniform degradation process of Equation 2, a kernel map is essential. Here, we present a strategy to generate a kernel map.

Before synthesizing kernel maps, we must discuss the characteristics of kernel maps from the wild. We chose three characteristics to consider in the process of kernel map generation. The three characteristics are: randomness, coherency, and incoherency.

Randomness. Kernels can be applied to random regions, regardless of the contents. A specific content is not constrained to certain kernels and any kind of kernel can be applied randomly. Coherency. Although kernels may be randomly applied to different regions, it is very likely that neighboring regions tend to have similar kernels applied. A kernel map should have spatial coherency. Incoherency. While it is very likely that kernels of each pixel tend to be similar, this is not always the case. The possibility that kernels can change abruptly at neighboring pixels still exists.

a: KERNEL COLLAGE

In order to assure that the kernels are coherent in neighboring pixels, but change abruptly at times, we propose a degradation method Kernel Collage. First, we generate $N$ kernel maps each filled with different kernels. One of these kernel maps is selected and is served as the base. The other $N - 1$ kernels are shuffled, cropped, and pasted onto the base kernel map, like a
Finally, once the Predictor is fully trained, the SR network is fine-tuned, so that it becomes robust to incorrect kernel estimations.

At test time, the LR image is first given to the Predictor and it makes an estimation on the kernel map. This estimated kernel map is given along with the LR image to the SR network (G), which outputs the final super-resolved image (I^{SR}).

C. OBJECTIVE FUNCTIONS

1) SUPER-RESOLUTION NETWORK

In SR network (G) training, a non-uniformly degraded LR image (I^{LR}) and the corresponding ground-truth kernel map (K̃) is given as the input. The network is trained to output a super-resolved image (I^{SR}) that is close to the ground-truth HR image (I^{GT}). Here, we use L1 distance as the objective function.

2) PREDICTOR

The Predictor (P) estimates the kernel map from the given LR image. We use an indirect method to train the Predictor, via SR reconstruction loss. The output kernel map from the Predictor (K̃) is given to the SR network (G) and the loss is computed from the output super-resolved image I^{SR}. The loss function can be formulated as the following:

$$\ell_c = |G(I^{LR}, K̃) - G(I^{LR}, \tilde{K})|_1$$ (3)

The simplest way to train the Predictor may be using the $L_p$ distance between the ground-truth kernel map (K) and the predicted kernel map (K̃) directly:

$$\ell_c = |K - \tilde{K}|_p^p$$ (4)

This may be effective in a uniform degradation scenario, but we find it to be unsuitable in non-uniform degradation scenarios. By using Equation 4 for training, the Predictor can get penalized at featureless regions where kernel estimations are extremely challenging to be done, such as regions of a cloudless sky and pattern-less walls. These regions can be penalized greatly although the kernel estimations are irrelevant from SR quality, hindering the overall training process. Considering that our ultimate goal is to super-resolve a given LR image, wrong estimations of kernels are acceptable as long as the corresponding parts are super-resolved correctly. For uniformly degraded LR images, this is not an issue and Equation 4 could work well since the network can obtain clues for kernel predictions from regions other than featureless regions. However, in a non-uniform circumstance, information from other regions is meaningless, since the kernels may differ. Hence, this problem becomes more difficult. Thus, in order to prevent penalizing regions irrelevant from SR quality, we train the Predictor via reconstruction loss. Here, we can make two choices on setting the target image of the super-resolved image. For one, we can simply use the ground-truth HR image (I^{GT}) as the target image:

$$\ell_c = |I^{GT} - G(I^{LR}, \tilde{K})|_1$$ (5)

B. OVERALL FRAMEWORK

By using the generated kernel map described in the previous section, synthetic LR-HR pairs can be obtained. Once the data pairs are prepared, the SR network (G) is trained first. For the SR network, we adopt SFTMD [4], which is known to utilize information on kernels well in the SR process. Since the original SFTMD architecture was designed for image-level kernel estimation, we slightly modify the architecture so that it is capable of utilizing pixel-level kernel estimation. Once the SR network is fully-trained, the Predictor (P), which estimates the kernel map from a given LR image, is trained.
Another option is to use the SR result from the SR network using the ground-truth kernel map \( \hat{K} \), as Equation 3. We choose the latter option (Equation 3) to train the Predictor. The reason for this is because by using Equation 3, the Predictor can focus more on kernel map estimation. If \( I^{GT} \) is set as the target, an erroneous loss irrelevant from estimations on the kernel map may flow to the Predictor. Specifically, the Predictor may get penalized even when it has made a correct estimation on the kernels, although the error is due to the limits of the main SR network \( G \). (See Figure 3 for a visualized explanation.)

![Figure 3](image-url)

**FIGURE 3.** Visualization of the difference maps between super-resolved results and the ground-truth image. If \( I^{GT} \) is used as the target image in training the Predictor, the Predictor may get penalized for the errors due to the SR network, even where the Predictor has made a correct estimation. Best on screen.

**D. FINE-TUNING**

Once the Predictor and the SR network are trained, the framework can be used immediately. However, using the Predictor and the SR network straightforwardly has a weakness. The predictions made by the Predictor may not be perfect; it can make some incorrect estimations at some pixels. On the other hand, the pre-trained SR network expects to get perfectly accurate kernel maps as the input along with the LR image. Consequently, it is prone to incorrect kernel estimations and may not perform fully well in practice. Therefore, the SR network needs to be fine-tuned so that it becomes robust to wrong estimations. With the weights of the Predictor fixed, the SR network is fine-tuned. In this process, the Predictor makes estimations on kernels and is given to the SR network along with the LR image, and the SR network is trained to recover the original GT image from it. Here, Equation 5 is used as the objective function for training the SR network.

**V. EXPERIMENTS**

**A. IMPLEMENTATION DETAILS**

For training, we use the HR images from DIV2K [27] and Flickr2K [28] dataset. We generate synthetic training image pairs via Kernel Collage. For simplicity, we consider isotropic and anisotropic Gaussian kernels in the experiments of this paper. The training data is augmented with random horizontal and vertical flips, along with random 90° rotations. We use a patch size of 64 for LR images and a batch size of 16. Our SR network and Predictor are trained from an initial learning rate of 2e-4 and 1e-4 respectively and are both reduced by a factor of 0.1 after 2 \( \times 10^5 \) iterations. We use Adam [29] for the optimizing process. The SR network and the Predictor are trained for 3 \( \times 10^5 \) iterations and 2.5 \( \times 10^5 \) iterations. The fine-tuning process takes 1 \( \times 10^5 \) iterations, with a learning rate starting from 1e-4, which is halved after every 2.5 \( \times 10^4 \) iterations.

1) **NETWORK ARCHITECTURES**

For our main super-resolution network \( (G) \), we have slightly modified SFTMD [4]. Using the original SFTMD model is not suitable for pixel-level kernel estimation, since it expects one kernel for an image. The \( \gamma \) and \( \beta \) values used in SFT-layers of the original SFTMD are single values, applied in a spatially uniform manner. Hence, we make the outputs of the SFT-layers be maps of \( \gamma \) and \( \beta \). We use four convolutional layers for processing the input kernel map and two more convolutional layers for computing \( \gamma \) and \( \beta \) respectively. We use LeakyReLU [30] \((\alpha = 0.2)\) for the activation functions. The predictor we used is stacked with 8 convolutional layers. The first 6 layers each use 64 \( \times 3 \) filters, while the last two use \( 1 \times 1 \) filters. The seventh layer use 64 filters and the final (eighth) layer uses 3 filters. All activations are LeakyReLU \((\alpha = 0.2)\) except the final one, which is the Sigmoid operation.

2) **EXPERIMENTS ON SYNTHETIC TEST IMAGES**

**Evaluation Datasets:** We evaluate our approach (PerPix) both quantitatively and qualitatively. Experiments in this paper are done at \( \times 4 \) scale unless mentioned, as most baseline methods provide pre-trained models and codes at this scale. Note that PerPix can be considered a deblurring network if the scale factor is set as \( \times 1 \), as in DAN [21]. PerPix is evaluated on two settings, both uniform and non-uniform degradation scenarios. First, we compare our approach to other methods on DIV2KRK [2]. It consists of 100 LR-HR image pairs, where each LR image is uniformly downsampled with different kernels.

Furthermore, since there is no adequate benchmark dataset for non-uniform degradation scenarios, we have synthesized datasets using 3 types of non-uniform degradation for evaluation:

1) Grid 2) Smooth 3) Grid-segmented smooth. 1) Grid degradation setting is where an image is divided into patches in a grid form and each patch is degraded uniformly. 2) Smooth degradation setting is where the kernels vary at each pixel, smoothly changing along space. 3) Grid-segmented smooth degradation setting is where kernels vary smoothly along space but change abruptly at certain points. The smoothly varying kernel map is segmented in a grid form as in 1). We apply 16, 64, and 144 different kernels for...
TABLE 1. Evaluation on DIV2KRK (Uniform degradation scenario). The best and second best performances are indicated with red and blue. Although our model is trained for non-uniform degradation scenarios, it also performs successfully in uniform degradation scenarios, achieving state-of-the-art performance both quantitatively and qualitatively. It even outperforms non-blind algorithms given the ground-truth (GT) kernels and algorithms based on repetitive corrections on kernel estimations. PerPix also shows an outstanding image quality, by a noticeable difference. The quantitative results besides $P + SFTMD$, IKC and PerPix (ours) are from [21] and results of $P + SFTMD$ and IKC are from the official implementation.

| Class | Method | PSNR | SSIM |
|-------|--------|------|------|
| 1     | Bicubic interpolation | 25.33 | 0.6795 |
|       | Bicubic kernel + ZSSR [17] | 25.61 | 0.6911 |
|       | EDSR [13] | 25.64 | 0.6928 |
|       | RCAN [15] | 25.66 | 0.6936 |
|       | PDN [24] - NTIRE’18 track4 | 26.34 | 0.7190 |
|       | WDSR [24] - NTIRE’18 track2 | 21.55 | 0.6841 |
|       | WDSR [24] - NTIRE’18 track3 | 21.54 | 0.7016 |
|       | WDSR [24] - NTIRE’18 track4 | 25.64 | 0.7144 |
|       | Ji et al. [25] - NTIRE’20 track1 | 25.43 | 0.6907 |
| 2     | Michaeli et al. [1] + SRMD [18] | 23.34 | 0.6530 |
|       | Michaeli et al. [1] + ZSSR [17] | 26.09 | 0.7138 |
|       | KernelGAN [2] + SRMD [18] | 25.71 | 0.7265 |
|       | KernelGAN [2] + USRNet [26] | 20.06 | 0.5359 |
|       | KernelGAN [2] + ZSSR [17] | 26.81 | 0.7316 |
|       | $P + SFTMD$ [4] | 26.73 | 0.7448 |
|       | IKC [4] | 27.36 | 0.7677 |
|       | DAN [21] | 27.55 | 0.7582 |
| 3     | GT kernel + SRMD [18] | 27.38 | 0.7655 |
|       | GT kernel + ZSSR [17] | 27.53 | 0.7446 |
| Ours  | PerPix | 27.69 | 0.7820 |

In addition to these, we have also synthesized two more versions of non-uniform degradation datasets: a) Instance-segmented and b) Instance-segmented smooth. These are similar to Grid degradation and Grid-segmented smooth degradation; the difference is that the non-uniformity appears according to the instance segmentation masks. We used the image data and labels from the panoptic segmentation task of COCO [31] to synthesize the datasets for this setting.

**Metrics:** For quantitative evaluation, we use three metrics. We adopt the conventional PSNR and SSIM [32] metrics (higher the better), along with a recently proposed Learned Perceptual Image Patch Similarity (LPIPS) [33] (lower the better) metric. LPIPS is a metric for estimating the perceptual similarity between a pair of images. PSNR and SSIM are evaluated on the Y channel of YCbCr and LPIPS is evaluated on RGB channels.

1) EXPERIMENTS ON UNIFORM DEGRADATION

We first evaluate our method on DIV2KRK, which is a benchmark for blind SR. Since the LR images from this dataset are downsampled uniformly, with one kernel per
image, it is a setting more optimal for image-level kernel estimation. We compare our method to methods of 4 types: Class 1) state-of-the-art SR models trained on bicubically downsampled images, Class 2) winners of the Blind-SR challenge of NTIRE’18 [24] and NTIRE’20 [34], Class 3) kernel estimation based blind SR algorithms, Class 4) Ground-truth kernel + non-blind SR methods.

Surprisingly, although our method is not designed for uniform degradation scenario, it outperforms methods designed for uniform degradation scenario and achieves state-of-the-art performance in PSNR and SSIM. PerPix outperforms IKC and DAN which are based on repetitive corrections on kernels, even though our method estimates kernels at one shot. Moreover, PerPix even outperforms non-blind algorithms given the ground-truth (GT) kernels, achieving remarkable results both quantitatively and qualitatively. Results on DIV2KRK can be found at Table 1.

2) EXPERIMENTS ON NON-UNIFORM DEGRADATION

In addition to uniform degradation setting, we also conduct evaluations on non-uniform degradation scenarios. As mentioned previously, we experiment on 9 different non-uniform degradation conditions (Table 2) and 2 non-uniform degradation conditions based on instance segmentation (Table 3). Our approach consistently shows the best performance compared to the previous methods by a large margin, in terms of all three metrics. Not only does our approach perform well on uniform degradation settings, but also performs excellently in non-uniform degradation settings.

C. ABLATION STUDY

1) PATCH-WISE KERNEL ESTIMATION AND SR

In this section, we compare our approach (pixel-level kernel estimation) to state-of-the-art methods conducted in a patch-wise manner. We first compare our approach with a method designed for patch-wise kernel estimation and SR [20], for direct comparison. As there are not enough methods explicitly designed for patch-wise kernel estimation, we also conduct an experiment to compare kernel estimation at image-level, patch-level, and pixel-level. We divide the non-uniformly degraded images into patches, where each patch is uniformly degraded, but non-uniform when combined as a whole. The patches are given to the state-of-the-art methods which assume uniform degradation.

a: DIRECT COMPARISON WITH METHOD DESIGNED FOR PATCH-WISE KERNEL ESTIMATION

To our knowledge, [20] is the only method designed explicitly for patch-wise kernel estimation and SR. As the official implementation of [20] only provide pre-trained weights for ×2 scale, we experiment at ×2 scale. For evaluation, we adopt the DIV2KRK benchmark for uniform degradation scenario and the Grid degradation (64) of our synthesized dataset for non-uniform degradation scenario. Experiments on both scenarios show the superiority of our approach compared to [20] (Table 4).

b: PATCH-WISE EXPERIMENT WITH METHODS FOR UNIFORM DEGRADATION

We have also conducted a study comparing PerPix (pixel-level kernel estimation) to state-of-the-art kernel estimation

| Method | Instance-segmented (IS) | IS Smooth |
|--------|-------------------------|-----------|
|        | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS |
| Bicubic | 23.23 | 0.5905 | 0.663 | 23.20 | 0.5852 | 0.669 |
| RCAN [15] | 23.39 | 0.6015 | 0.626 | 23.35 | 0.5980 | 0.639 |
| P + SFTMD [4] | 23.05 | 0.6189 | 0.546 | 23.08 | 0.6195 | 0.555 |
| IKC [4] | 23.49 | 0.6136 | 0.594 | 23.65 | 0.6128 | 0.604 |
| DAN [21] | 23.53 | 0.6076 | 0.614 | 23.48 | 0.6043 | 0.625 |
| PerPix (Ours) | 26.31 | 0.7209 | 0.412 | 26.14 | 0.7143 | 0.427 |
FIGURE 4. Qualitative result on a real-world image from [35]. Our approach shows the best perceptual quality with minimum artifacts. Our method has a sharper image quality than results of RCAN and DAN and has less artifacts than IKC.

TABLE 4. Evaluation on uniform degradation(DIV2KRK) and non-uniform degradation(Grid-64) with comparison to [20] at ×2 scale. Our approach shows improvement by a noticeable margin.

| Method               | DIV2KRK PSNR | DIV2KRK SSIM | Grid (64) PSNR | Grid (64) SSIM |
|----------------------|--------------|--------------|----------------|----------------|
| Cornillere et al. [20] | 29.42        | 0.8549       | 27.58          | 0.8089         |
| PerPix (ours)        | 31.00        | 0.8883       | 31.86          | 0.9118         |

TABLE 5. Evaluation on Grid degradation setting (64). Divide-and-conquer form of approach at patch-level turns out to be unsuccessful and rather shows a better performance when done at image-level.

| Method         | PSNR | SSIM | LPIPS |
|----------------|------|------|-------|
| IKC (Patch-level) [4] | 19.10 | 0.5353 | 0.497 |
| IKC (Image-level) [4] | 25.20 | 0.6912 | 0.506 |
| DAN (Patch-level) [21] | 19.53 | 0.5480 | 0.544 |
| DAN (Image-level) [21] | 25.26 | 0.6812 | 0.531 |
| PerPix (Pixel-level) | 28.73 | 0.7996 | 0.339 |

TABLE 6. Ablation study on the indirect loss. Use of our proposed indirect loss consistently shows an improvement in SR at all metrics.

| Loss option                  | PSNR | SSIM | LPIPS |
|------------------------------|------|------|-------|
| Direct loss (Eq. 4)          | 23.67 | 0.7020 | 0.388 |
| Loss with GT (Eq. 5)         | 27.00 | 0.7642 | 0.383 |
| Indirect loss (Eq. 3) (ours) | 27.03 | 0.7675 | 0.369 |

TABLE 7. Evaluation on the effect of fine-tuning. The fine-tuning phase makes the SR network get robust to a few incorrect kernel estimations.

|                     | PSNR | SSIM | LPIPS |
|---------------------|------|------|-------|
| Before fine-tuning  | 27.03 | 0.7675 | 0.369 |
| After fine-tuning   | 27.69 | 0.7820 | 0.352 |

and SR methods (IKC [4] and DAN [21]) conducted in a patch-wise manner. We report the performance on Grid degradation (64) of our synthetic non-uniform dataset. For the dividing process of images to patches, we divide the patches along the borders of the grid — where the kernels change abruptly when synthesizing LR data.

Although the problem of super-resolving a non-uniformly degraded image has transformed into a problem of
FIGURE 7. Illustration of Grid degradation (16). An example kernel map of Grid degradation (16) is visualized. A kernel map is divided into 16 different regions, and each region is uniformly filled with a single type of kernel. It can be seen from both magnified examples that the kernels within an identical region are filled with the same kernels, while changing abruptly, without coherency at the borders of the regions. Variations can be made depending on the number of regions to divide into — Grid degradation (64, 144). The regions are split to $8 \times 8$ and $12 \times 12$ respectively, in our experiments. In the experiment of Section 5.2.1, we have done an ablation study with comparison to patch-wise kernel estimation and SR. In order to conduct kernel estimation and SR in a patch-wise manner, we have divided the images along the borders of the grid (red line). This way, each patch is degraded with one kernel, which makes it a uniform degradation scenario. Thus, the problem of kernel estimation and SR on non-uniformly degraded LR transforms into a problem of multiple kernel estimation and SR on uniformly degraded LR images. Of all the non-uniform degradation scenarios mentioned in this work, this is the setting where comparison between patch-wise kernel estimation and pixel-level kernel estimation becomes fair the most. As reported in Section 5.2.1, although the non-uniform degradation scenario has transformed into a number of uniform degradation scenarios, which is the condition that models designed for uniform degradation scenarios should perform to its best, it rather performs worse than conducting kernel estimation and SR at image-level, estimating one kernel for an image.

FIGURE 8. Illustration of Smooth degradation (16). An example kernel map of Smooth degradation (16) is visualized. The kernels of the entire kernel map change smoothly, without any inconsistency. The process of generating a smoothly varying kernel map is shown at Figure 9.

super-resolving several uniformly degraded images, patch-wise kernel estimation and SR fails in practice, performing even worse than image-level kernel estimation and SR. The main reason for this performance drop is suspected to be due to the small size of patches. The dividing process makes the image smaller, making the prediction on kernels be made with much less information. This makes the process of kernel estimation and SR more difficult than the original state.
2) STUDY ON INDIRECT LOSS
To verify the effectiveness of our proposed indirect loss, we have compared it with Equation 4, 5. We perform two experiments to validate our approach. First, we conduct SR with a pre-trained SR network and the Predictors trained with each loss. The estimated kernel maps of the Predictor are

FIGURE 9. Illustration of generating a smoothly varying kernel map. This is an example of Smooth degradation (16), where 16 kernels are used in this process. 16 kernels are randomly chosen and positioned first, forming an initial kernel map. This is stretched to the resolution of the LR image, via interpolation. More variations can be made by configuring different number of kernels to use in this process — we have experimented with 64 and 144 in this paper.

FIGURE 10. Illustration of Grid-segmented smooth degradation (16). An example kernel map of Grid-segmented Smooth degradation (16) is visualized. A kernel map is divided into 16 different areas in a grid form, where each areas are filled with smoothly-varying kernels. As can be seen at the rightmost upper image, kernels within an identical region change smoothly, while this coherency is lost when it comes to the borders of the regions. (rightmost bottom image).
FIGURE 11. Illustration of instance-segmented degradation. Using the segmentation labels provided from the COCO dataset [31], each region labeled with the same segmentation masks are applied with the same kernels. The incoherency of kernels can be observed at the borders of each instances. Instance-segmented smooth degradation is somewhat similar to this, where the same regions (regions of the same segmentation masks) are applied with smoothly varying kernels; different from this example, where each instance is applied with one type of kernel.

FIGURE 12. Example LR images generated using the instance segmentation masks.

given to the pre-trained SR network and the SR results are evaluated. Note that we do not use the SR network fine-tuned after training the Predictor for fair comparison on pixel-level kernel estimation ability. We use the DIV2KRK dataset for this experiment and our proposed loss shows the best performance (Table 6).

Additionally, we conduct an experiment on computing the mean squared error (MSE) of the estimated kernels of each pixel. Since pixel-level GT kernel maps are necessary for this experiment, we use our synthesized non-uniformly degraded dataset. We report the results on Grid-Segmented (GS) smooth data (16, 64, 144). Our proposed loss shows the least error in kernel estimation, while the Predictor trained with Eq.5 shows the worst performance, although it showed an acceptable performance in experiments using SR. The Predictor trained with Eq. 4 shows a less error than that, which is because the Predictor simply tends to estimate an average kernel in order to minimize the loss. Based on these two experiments, the effectiveness of our proposed indirect loss can be confirmed.

3) STUDY ON THE EFFECT OF FINE-TUNING

Since the pre-trained SR network expects the given kernel maps to be accurate, it is prone to incorrect estimations at some pixels. Therefore, the SR network must get trained to be robust to such incorrect estimations. Fine-tuning shows an improvement in all metrics. (Table 7).
FIGURE 13. Qualitative results on Grid degradation (16, 64, 144). At all three settings, PerPix performs stably, showing a consistently excellent image quality compared to other state-of-the-art models. Image-level kernel estimation based methods like IKC and DAN inevitably either over-sharpens some regions or blurs some regions in non-uniform degradation scenarios, while PerPix do not show such artifacts.

D. EXPERIMENT ON REAL IMAGES
Besides the experiments on synthetic test images, we also conduct an experiment on real-world images. We use images from [7] and [35] and compare it to state-of-the-art SR models. Qualitative results can be found at Figures 4 and 5.

E. INFERENCE SPEED
We compare the inference time with blind-SR models. IKC [4] and DAN [21] takes 2.81s and 0.52s, respectively, while PerPix takes only 0.09s per image in average. This difference in inference time is mainly due to the absence of the iterative loop in PerPix. The timings are measured with the DIV2KRK dataset using a single RTX Titan GPU.

F. PROCESSING TIME OF KERNEL COLLAGE
Uniform degradation applies identical degradation kernels throughout an image, enabling parallelization of the degradation process thanks to convolution operation using modern graphics processing units (GPUs). On the other hand, when it comes to non-uniform degradation, each region is degraded with different kernels, making it difficult to be parallelized. Kernel Collage is a non-uniform degradation method, applying different kernels at different regions. Therefore, one may
think that Kernel Collage would take a vast amount of time for computation. However, Kernel Collage has merit in that the degradation process can be parallelized. As explained in Section IV-A1, identical kernels can be applied at neighboring regions, enabling parallelization and the use of GPUs for processing.

In specific, we have measured the processing time of Kernel Collage with comparison to the conventional uniform degradation. We used the DF2K dataset with a single RTX Titan GPU for this measurement. It took 0.0024 seconds per batch on average to process uniform degradation (1 degradation kernel per image), while kernel collage with 4 degradation kernels and 16 degradation kernels took 0.0030 seconds and 0.0052 seconds, respectively. We strongly argue that this difference in processing time is negligible, since our proposed model PerPix takes 0.0577 seconds per batch on average, occupying the majority of training time. Note that we found using 4 degradation kernels per image to perform best, which only increases the training time by less than 1%.

VI. CONCLUSION
We have shown that image-level kernel estimation and patch-level kernel estimation have limits in handling the
non-uniformity of kernels and have demonstrated that pixel-level estimations on kernels can exceed these limits. We have also introduced PerPix, a SR framework based on pixel-level kernel estimation. To train this framework, we used Kernel Collage, a simple yet effective method to synthesize non-uniform degradation and the indirect loss to train pixel-level kernel estimation. Our approach outperforms previous blind-SR algorithms by a large margin quantitatively and qualitatively, on both uniform and non-uniform degradation datasets.

APPENDIX A

NON-UNIORMLY DEGRADED DATASETS

Since there is no adequate benchmark dataset for non-uniform degradation scenarios, we have synthesized 11 datasets to evaluate our approach. Five types of degradations are used in total: 1) Grid 2) Smooth 3) Grid-segmented smooth 4) Instance-segmented 5) Instance-segmented Smooth Degradations 1), 2), and 3) can be done in diverse variations. Illustrations describing these can be found at Figures 7 ~ 12.
FIGURE 16. Additional qualitative results on DIV2KRK benchmark (Uniform degradation scenario). PerPix outperforms the state-of-the-art blind-SR algorithms designed for uniform degradation scenarios.

APPENDIX B
ADDITIONAL QUALITATIVE RESULTS
More qualitative results on our synthesized LR images and DIV2KRK can be found at Figures 13 ~ 16.

REFERENCES
[1] T. Michaeli and M. Irani, “Nonparametric blind super-resolution,” in Proc. IEEE Int. Conf. Comput. Vis., Dec. 2013, pp. 945–952.
[2] S. Bell-Kligler, A. Shocher, and M. Irani, “Blind super-resolution kernel estimation using an internal-GAN,” in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 284–293.
[3] N. Efrat, D. Glasner, A. Apartsin, B. Nadler, and A. Levin, “Accurate blur models vs. image priors in single image super-resolution,” in Proc. IEEE Int. Conf. Comput. Vis., Dec. 2013, pp. 2832–2839.
[4] J. Gu, H. Lu, W. Zuo, and C. Dong, “Blind super-resolution with iterative kernel correction,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 1604–1613.
[5] S. Harmeling, H. Michael, and B. Schölkopf, “Space-variant single-image blind deconvolution for removing camera shake,” in Proc. Adv. Neural Inf. Process. Syst., vol. 23, 2010, pp. 829–837.
[6] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, “Non-uniform deblurring for shaken images,” Int. J. Comput. Vis., vol. 98, no. 2, pp. 168–186, 2012.
[7] J. Cai, H. Zeng, H. Yong, Z. Cao, and L. Zhang, “Toward real-world single image super-resolution: A new benchmark and a new model,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 3086–3095.
[8] J. Sun, W. Cao, Z. Xu, and J. Ponce, “Learning a convolutional neural network for non-uniform motion blur removal,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 769–777.
Eunyeon Kim received the B.S. degree in computer science and engineering from Sungkyunkwan University (SKKU), Suwon, South Korea, in 2021, where she is currently pursuing the master’s degree in artificial intelligence. Her research interests include implicit neural networks, generative neural networks, information theory, machine learning, and deep learning.

Jae-Pil Heo (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer science from the Korea Advanced Institute of Science and Technology (KAIST), in 2008, 2010, and 2015, respectively. He is currently an Assistant Professor with Sungkyunkwan University (SKKU), South Korea. Before joining SKKU, he was a Researcher with the Electronics and Telecommunications Research Institute (ETRI). His research interests include computer vision and machine learning.