Unsupervised Learning for Real-World Super-Resolution

Andreas Lugmayr  Martin Danelljan  Radu Timofte
CVL, ETH Zürich, Switzerland

Abstract

Most current super-resolution methods rely on low and high resolution image pairs to train a network in a fully supervised manner. However, such image pairs are not available in real-world applications. Instead of directly addressing this problem, most works employ the popular bicubic downsampling strategy to artificially generate a corresponding low resolution image. Unfortunately, this strategy introduces significant artifacts, removing natural sensor noise and other real-world characteristics. Super-resolution networks trained on such bicubic images therefore struggle to generalize to natural images.

In this work, we propose an unsupervised approach for image super-resolution. Given only unpaired data, we learn to invert the effects of bicubic downsampling in order to restore the natural image characteristics present in the data. This allows us to generate realistic image pairs, faithfully reflecting the distribution of real-world images. Our super-resolution network can therefore be trained with direct pixel-wise supervision in the high resolution domain, while robustly generalizing to real input. We demonstrate the effectiveness of our approach in quantitative and qualitative experiments.

1. Introduction

Super-resolution (SR) aims to enhance the resolution of natural images. Recent years have seen an increased interest in the problem, driven by emerging applications. Most notably, current generations of smartphones allow for the deployment of powerful image enhancement techniques, based on machine learning approaches. This calls for super-resolution methods that can be applied to natural images, that are often subject to significant levels of sensor noise, compression artifacts or other corruptions encountered in applications. In this work, we therefore address the problem of super-resolution in the real-world setting.

Real-world SR poses a fundamental challenge that has been largely ignored until very recently. The lack of natural low resolution (LR) and high resolution (HR) image pairs greatly complicates the evaluation and training of SR methods. Therefore, research in the field has long relied on the use of known degradation operators such as bicubic kernel in order to artificially generate a corresponding LR image [8, 29, 31]. While this straightforward approach enables simple and efficient benchmarking and generation of virtually unlimited training data, it comes with significant drawbacks. Bicubic downsampling can drastically change the natural characteristics of an image by, e.g., removing sensor noise and compression artifacts.

State-of-the-art methods trained only to reconstruct images artificially downsampled with a bicubic kernel, do not generalize to natural images. As visualized in Figure 1, even small levels of noise causes a network trained only on bicubic images, in this case ESRGAN [32], to output significant artifacts. In fact, this is expected as deep learning methods are known to be sensitive to significant differences
between the train and test distributions. The ESRGAN has not seen noisy input images during train-time due to the smoothing effects introduced by bicubic downsampling.

In this work, we present a novel way of training a generic method in order to overcome the challenges of real-world SR. We address the shift between training and testing distributions arising from the bicubic downsampling by learning the corresponding inverse mapping operation. To this end, we train a mapping from the bicubic images to the distribution of real-world LR images. By employing cycle consistency losses [37], we learn this mapping in a fully unsupervised manner. The learned network is applied on bicubically downsampled images to generate paired LR and HR images that follow the real-world distribution. This allows us to learn the SR network on a realistic dataset, unaffected by the bicubic shift. Furthermore, the SR network is trained with direct pixel-wise supervision in the HR domain, without the need of any paired ground-truth data. Visual results of our approach on natural images are shown in Figure 1.

Due to the unavailability of paired data, we introduce a protocol for benchmarking real-world SR methods, based on simulating natural degradations. We analyze our approach in two scenarios, namely Domain (DSR) and Clean Super-Resolution (CSR). In the former case, the real-world data distribution is defined by one set of natural images. However, our approach generalizes to the case when the real-world input and output distributions of the SR network are different. We therefore introduce the CSR task, where the goal is to achieve a clean super-resolved image, defined by a separate output distribution of high-quality images. We demonstrate the effectiveness of our approach on the aforementioned benchmark, and compare it to baseline methods and state-of-the-art approaches. Finally, we show qualitative results for the task of super-resolving real-world smartphone images on the DPED [16] dataset.

2. Related Work

Until very recently, single image super-resolution (SISR) methods were primarily benchmarked in terms of PSNR, for the task of super-resolving bicubic downscaled images. While traditionally addressed with classical techniques [18, 11, 27, 33, 14], current approaches [8, 9, 21, 22, 25, 10, 1, 2, 13, 15] employ deep learning methodologies to train a mapping from LR to HR. Among the latter, EDSR [25] notably introduced a ResNet inspired architecture, better adapted for the task at hand. For training the network however, these methods rely on the L1 or L2 losses. While these losses are closely related to the PSNR evaluation metric, they do not preserve the natural image characteristics, generally leading to a blurry result [35]. To address this problem, Ledig et al. [23] introduced an objective function aimed at perceptually more pleasing results. The novel objectives were a GAN discriminator and a loss computed in the VGG feature space. While providing inferior PSNR compared to state-of-the-art, the super-resolved images experienced significantly better perceptual quality. Following this philosophy, the recent winner of the PIRM2018 [17] challenge ESRGAN [32], proposed further architectural improvements to further enhance the perceptual quality.

Despite their success, the aforementioned approaches are severely limited by their reliance on the bicubic downsampling operation for training data generation. This operation eliminates most high frequency components and therefore, significantly altering the natural image characteristics, such as noise, compression artifacts, and other corruptions. The bicubic assumption therefore rarely reflects the real-world scenario. Blind SR generalizes the problem by assuming LR and HR image pairs with an unknown degradation and downsampling kernel. Early attempts [3] to this problem include explicitly estimating the unknown point spread function itself [26, 12]. Another direction of research aims to completely remove the need for external training data by performing image-specific SR. Following this idea, ZSSR [28] trains a lightweight network using only the testing image itself, by performing extensive data augmentation. However, this approach still employs a fix downsampling operation to generate synthetic pairs at test time. Furthermore, the image-specific learning leads to extremely slow prediction.

A few recent works address the unsupervised SR setting, where no paired LR-HR pairs are given and the relation between LR and HR images is unknown. The Cycle-in-Cycle network [34] learns a mapping from the original input image to a clean image space, using a framework that employs cycle consistency losses. The SR network itself is trained by only employing indirect supervision in the LR domain, in addition to the usual perceptual GAN-discriminator. In contrast, our framework allows direct supervision in the HR domain, resulting in better training of the SR network itself. Furthermore, instead of “cleaning” the input image during train and test time, we learn a mapping to the original input domain for only the training. Another work focuses on the downsampling process [20] in order to improve the SR. However, SR is only performed on images with the learned downsampling operation, and is therefore not applicable to our real-world scenario. Also Bulat et al. [4] focus on the problem of learning the downsampling process. However, this approach specifically addresses the problem of super-resolving faces, where strong content priors can be learned by the network. In contrast, we tackle the general SR problem, not putting any assumptions on the image content. Lastly, recent works [36, 6, 5] propose strategies to capture real LR-HR image pairs. However, these methods rely on complicated data collection procedures, requiring specialized hardware, that is difficult and expensive to scale. Our approach operates without the need of any additional data,
3. Proposed Method

3.1. The Super-Resolution Problem

In essence, super-resolution (SR) is the problem of increasing the resolution of natural images. However, this problem comes with a fundamental challenge that has been largely ignored up until very recently. Namely, the lack of natural LR and HR image pairs, which are needed for evaluation and training. Therefore, research in SR has long relied on the use of known downscaling operators (e.g., bicubic) in order to artificially generate a corresponding LR image pair. While this simplification has historically also served the development of SR methods, it is fundamentally limiting.

Bicubic downsampling can drastically change the natural characteristics of an image by, e.g., removing sensor noise and compression artefacts. A real-world example is shown in Figure 2. The natural image (left) is affected by natural sensor noise. However, the corresponding bicubically downsampled image does not preserve these characteristics. Hence, a network trained to super-resolve the latter image cannot be expected to generalize to the original real-world distribution.

To formalize the problem, we let $X$ denote the natural image we wish to super-resolve. We also introduce the distribution $p_X$ of such natural images $X \sim p_X$ on which we want our SR approach to operate. In practice, $p_X$ could be defined as images obtained from a specific camera or a dataset of real-world images. The aim is to learn a function $S$ that maps an image $X \sim p_X$ to a high resolution image $\hat{Y} = S(X)$ that is distributed according to the output distribution $p_{\hat{Y}}$. In applications, we could have $p_{\hat{Y}} = p_Y$, meaning that we want the characteristics of the image to remain unchanged after super-resolution. We term this setting domain-specific super-resolution (DSR). Another alternative would be to let $p_{\hat{Y}}$ be defined by a set of high-quality images, which we call clean super-resolution (CSR) setting.

For most real-world applications it is incredibly hard and strenuous to collect natural image pairs $(X,Y)$ for SR. In classical SR this is addressed by artificially constructing the input image $Z = B(Y)$, where $B$ is the bicubic downsample operation. The task is then aimed to super-resolve $Z$ to match the original image $S(Z) \approx Y$. However, as illustrated in Figure 2, the bicubically downsampled images $Z \sim p_Z$ do not match the input distribution, i.e., $p_Z \neq p_X$. Unfortunately, methods trained in this manner struggle when supplied with real data $X \sim p_X$.

Related to our discussion is the concept of blind SR. In this setting, the input images $X$ are assumed to be generated from the output images $Y$ with some fixed and simple transformation that is unknown. Often, a more general downsampling kernel $k$ is used in combination with a non-linear degradation function $f_{\text{deg}}$, such that $X = f_{\text{deg}}(k * Y)$. Some methods try to find the kernel $k$ from data or learn the transformation end-to-end.

3.2. Overview

The real-world SR setting, addressed in this work, can be seen as a generalization of blind SR. In our approach, we assume no particular relation, such as a parameterized transformation, between the input $X$ and output $Y$ images. We only assume that a set of input image samples $\{X_i\}_{i=1}^M \sim p_X$ and a set of output image samples $\{Y_i\}_{i=1}^N \sim p_Y$ are available. These image samples are not paired. Given this data, the problem is to learn a mapping $S$ that can super-resolve a new image $X \sim p_X$ such that $S(X) \sim p_Y$. In order to train $S$ from such unpaired data, we learn a function $\hat{X} = G(Z)$ that maps the bicubically downsampled image $Z = B(Y)$ from the output distribution to an image sample $\hat{X} \sim p_X$ that fits the input distribution $p_X$. This effectively constructs an input-output training pair $(\hat{X}, Y)$, allowing the SR network $S$ to be learned in a supervised manner such that $S(\hat{X}) \approx Y$. The main advantage of our approach is that the SR network can be trained with direct pixel-wise supervision in the HR domain. The proposed framework is depicted in Figure 3.

We first train the generator $G$, called the domain distribution network, in a conditional GAN setting. This is performed by employing a discriminator network aiming to differentiate the generated images $\hat{X} = G(B(Y)), Y \sim p_Y$ from true input images $X \sim p_X$. Since no paired output is available, we enforce a cycle consistency loss by employing a second generator $F$ mapping input images $X$ to $\hat{Y} = F(X) \sim p_Z$. Crucially, we train the domain distribution network $G$ independently from the SR network $S$. While, this may seem counter intuitive at first, it is clearly motivated from the fact that the networks $G$ and $S$ have fundamentally conflicting objectives. The aim of $G$ is to map a bicubically downsampled image $Z = B(Y)$ from the output distribution to an image $\hat{X}$ following the input distribution $p_X$, such that a faithful training sample $(\hat{X}, Y)$ is generated for the SR network. The network $S$ simply
aims to super-resolve any image from \( p_X \). If both networks were to be trained jointly using the cycle-consistency loss for \( S(G(B(Y))) \approx Y \), the networks \( S \) and \( G \) would collaborate in order to minimize the aforementioned loss. This leads to severe overfitting and poor generalization. As illustrated in Figure 3, we train the SR network is a second separate training stage, using the training pairs generated by the network \( G \).

### 3.3. Domain Distribution Learning

The task of the domain distribution learning \( \hat{X} = G(Z) \) is to map a bicubic downsampled image from the output distribution \( Z = B(Y) \) to the input distribution \( p_X \). Since we do not have access to paired samples, we need to venture into unsupervised learning territories. We firstly employ a GAN discriminator \( D_X \), tasked to differentiate between the generated \( G(Z) \) and images drawn from the input distribution \( p_X \). For this, we employ the original GAN formulation,

\[
\mathcal{L}_{GAN}(G, D_X) = \mathbb{E}_{X \sim p_X} [\log D_X (X)] + \mathbb{E}_{Y \sim p_Y} [\log(1 - D_X (G(B(Y)))).
\]

(1)

To preserve the image content, despite the lack of paired images, we employ cycle consistency losses [7]. A second generator \( F \) is tasked to map images from the input domain \( p_X \) to the domain of bicubic downsampled images \( p_Z \), where \( Z = B(Y) \). We then add cycle consistency losses as,

\[
\mathcal{L}_{cyc}(X, Y) = \mathbb{E}_{Y \sim p_Y} \left[ ||F(G(B(Y))) - B(Y)||_1 \right] + \mathbb{E}_{X \sim p_X} \left[ ||G(F(X)) - X||_1 \right].
\]

(2)

They constrain the generators \( G \) and \( F \) to be each others approximate inverses. Hence, the image shall be preserved if mapped through \( G \) and then \( F \) back to the original domain. Analogous to (2), we add a discriminator \( D_Z \) and similar loss on the bicubic side. The full objective is thus,

\[
\mathcal{L}_{DDL}(G, F, D_X, D_Y) = \mathcal{L}_{GAN}(G, D_X) + \mathcal{L}_{GAN}(F, D_Z) + \lambda \mathcal{L}_{cyc}(G, F).
\]

(3)

The full architecture is shown in Figure 3 (blue).

**Network architectures** For our experiments we designed the domain distribution mapping \( G \) based on the CycleGAN architecture [7]. The generators \( G \) and \( F \) use a ResNet architecture with nine blocks. We replace the transposed convolution layers with bi-linear upsampling followed by a standard convolution. We found this to be beneficial for learning stability, and it effectively removed checkerboard pattern artifacts. Furthermore, we found the \( \tanh \) non-linearity on the output to be harmful for color consistency, and therefore use no non-linear activation at the output. The discriminators \( D_X \) and \( D_Y \) consist of a three-layer network architecture that operate on a patch level [24, 19].

**Training details** We adopt the training procedure proposed in CycleGAN, using 200 epochs and the Adam optimizer with \( \beta_1 = 0.5 \). The starting learning rate is set to 0.0002.

### 3.4. Super-Resolution Learning

Here we describe the learning of the SR network \( S \). In the absence of paired ground-truth data, we train the network with pairs \((\hat{X}_j, Y_j)\), where the input image \( \hat{X}_j = G(B(Y_j)) \) is generated by our domain distribution network \( G \). We employ the pixel-wise content loss [25],

\[
\mathcal{L}_1(S) = \mathbb{E}_{Y \sim p_Y} ||S(\hat{X}) - Y||_1.
\]

(4)

Following the success of SRGAN [23], we also employ the VGG feature loss, that is known to better correlate with per-
ceptual quality

\[ L_{VGG}(S) = \mathbb{E}_{Y \sim p_Y} \| \phi(S(\hat{X})) - \phi(Y) \|^2_2. \]  

(5)

Here, \( \phi \) denotes the feature activations extracted from the VGG network. We extract the features at the same depth as SRGAN, which is after the activation of the 4th convolutional layer, before the 5th maxpooling layer.

For better perceptual quality, we further employ a GAN discriminator \( D_Y \). To this end, we adopt the relativistic discriminator employed in ESRGAN \[32\]. As opposed to the conventional discriminator, providing an absolute real/fake probability for each image, a relative score real/fake is estimated compared to a set of real of fake images.

\[ D_Y(Y, \hat{Y})(C) = \sigma(C(Y) - \mathbb{E}[C(\hat{Y})]) \]  

(6)

Where \( C(Y) \) is the raw discriminator output and \( \sigma \) is the sigmoid function. The SR network is trained with added perceptual loss,

\[ L_{RaGAN}(S, C) = -\mathbb{E}_{Y \sim p_Y} \log(1 - D_Y(Y, S(X))) \]  

\[ -\mathbb{E}_{X \sim p_X} \log(D_Y(S(X), Y)) \].  

(7)

This results in the total loss of

\[ L(S, D) = L_{VGG}(S) + \lambda L_{RaGAN}(S, C) + \eta L_1(S). \]  

(8)

The GAN loss is multiplied by a weight \( \lambda \), balancing the guidance of the two pixel-wise losses \( L_{VGG} \) and \( L_{RaGAN} \) against the GAN loss \( L_{GAN} \).

**Network architecture** Our approach is agnostic to the specific architecture of the SR network \( S \). For simplicity, we adopt the recently proposed ESRGAN architecture, which is the winner of the PIRM 2018 challenge \[17\]. It introduced a new building block called Residual-in-Residual Dense Blocks, improving stability of training. We augment the ESRGAN network with a final color adjustment layer, to ensure a faithful reproduction of the color palette in the input LR image. This layer adjusts the local mean RGB value to that of the low-resolution image.

**Training details** To train our SR network, we start from pre-trained ESRGAN \[32\] generator and discriminator networks. We then perform 50000 training iterations. We use that ADAM optimizer with an initial learning rate of \( 10^{-4} \) and set \( \beta_1 = 0.9 \) and \( \beta_2 = 0.999 \) for both the Generator and Discriminator. We use the learning rate schedule in \[32\], decreasing it by a factor of 0.5 after 10%, 20%, 40% and 60% of the total number of iterations.

**4. Experiments**

In this section, we present comprehensive quantitative and qualitative evaluation of our approach. We first discuss the setup and datasets employed in our experiments. Detailed results are provided in the supplementary material.

---

**Figure 4.** Overview of our data generation procedure for benchmarking unsupervised SR methods. See text for details.

**4.1. Experimental Setup**

We present a novel strategy for evaluating real-world SR methods. In traditional SR the bicubic downsampled image \( Z = B(Y) \) is super-resolved and compared to \( Y \). In the real-world scenario, we do not have access to a ground-truth image, complicating quantitative analysis. On the other hand, we can closely simulate the real-world scenario by constructing the sets input \( \{X_i\}_{i=1}^M \) and output \( \{Y_j\}_{j=1}^N \sim p_Y \) training images by applying downsampling and synthetic degradations to a dataset of original images. The type of degradation is unknown to the SR approach.

For evaluation, we further generate a set of ground truth pairs \( (\hat{X}_i, \hat{Y}_i) \), where \( \hat{X}_i \sim p_X \) and \( \hat{Y}_i \sim p_Y \). These are inaccessible to the network during training, and only used for evaluation purposes. We consider two scenarios: DSR and CSR, detailed below.

**Domain-Specific Super-Resolution (DSR)** The input \( X \) and target \( Y \) images shares the same real-world distribution, i.e. \( p_X = p_Y \). Thus, the aim is to produce super-resolved images \( \hat{Y} \) that are of the same distribution as its input images. The training set is generated by first downsampling the image and then simulating the real-world degradation. In the DSR case the same training set \( \{X_i\}_{i=1}^M \) represents both the input \( p_X \) and output \( p_Y \) distribution. For evaluation, the input image \( \hat{X}_i \) is constructed using the same procedure as for the training images \( X_i \). The corresponding ground-truth output image \( \hat{Y}_i \) is obtained by directly adding the degradation to the original HR image. The procedure is visualized in Figure 4.

In **Clean Super-Resolution (CSR)**, the goal is to super-resolve an input image \( X \) such that the output image \( \hat{Y} \) fits another distribution \( p_Y \). We let \( p_Y \) be defined by a dataset of high quality images. Therefore, we employ the unaltered original image from the dataset to be the ground-truth output \( \hat{Y}_i \). The corresponding LR image \( \hat{X}_i \) used for the evaluation is generated as in the DSR case above. We also employ the same training set of input images \( \{X_i\}_{i=1}^M \) as for the DSR. In the CSR case however, the output training data \( \{Y_j\}_{j=1}^N \) represent a different distribution of clean images.
PSNR = 20.81
LPIS = 0.7889

PSNR = \infty
LPIS = 0

PSNR = 18.90
LPIS = 0.7837

EDSR  GT  Ours

Figure 5. While the PSNR is only dependent on the pixel-wise distance between GT and the prediction, LPIPS is a more elaborate measure that takes the perceptual quality into account. Although the EDSR image is perceptually much worse than the prediction of our method, it scores higher in PSNR. The LPIPS distance is 56% smaller for our method, which is perceptually superior.

These are generated by bicubically downsampling the original image. The resulting image thus represents a clean ideal output from the SR network. See Figure 4 for a schematic description of the procedure.

**Degradations** To model the real-world setting we evaluate unsupervised SR approaches using two types of image degradations: JPEG compression artifacts and simulated sensor noise. In case of JPEG artifacts we use a quality setting of 30. JPEG compression artifacts are a common when applying super-resolution to images captured by smartphones or acquired from the internet. In the second case, we employ white Gaussian noise with a standard deviation of $\sigma = 8$. This simulates the case of real-world sensor noise present in e.g., low light conditions or small sensor sizes.

**Quantitative Evaluation Measures** In order to quantitatively compare the different approaches we use the distance metrics PSNR, SSIM and LPIPS. While PSNR and SSIM are handcrafted methods, LPIPS is a learned metric for perceptual similarity [35] between two images. In Figure 5 we provide a comparison of PSNR and the LPIPS, measures using the model provided by the authors.

**Datasets** We use the DF2K [32] dataset that was introduced for learning the ESRGAN. It is a merge of the DIV2K [30] with 800 and the Flickr2K [30] dataset with 2640 images. The mean size of DF2K is 1439x1935. We also perform experiments on the DPED [16] dataset, acquired by a smartphone camera. It contains natural images with real-world sensor noise and other effects.

## 4.2. Ablation Study

For our ablation study we use the DF2K dataset as training data and the validation image from DIV2K to measure the performance of the different methods. The quantitative comparison is done using the PSNR, SSIM and LPIPS measures. For comparisons, we mainly consider the LPIPS distance due to its higher correlation with perceptual similarity.

We evaluate four different approaches for the DSR and CSR settings. All methods are trained using the same settings. For SR network we employ a pretrained ESRGAN model that is then fine-tuned for each method, as described below. Quantitative and visual results are shown in Table 1 and Figure 6 respectively.

**Baseline** First, we compare with the standard approach of training the network on LR images generated by bicubic downsampling. For this purpose, we finetune the ESRGAN using image pairs $(Z, Y)$. In the case of sensor noise, the baseline achieves significantly inferior performance for both DSR and CSR (Table 1) compared to ours. This is due to the smoothing behaviour of the bicubic downsampling. The baseline ESRGAN does thus not see appropriate levels of noise during training. This leads to severe artifacts in both the DSR and CSR case (Figure 6). Our approach also improves in the case of JPEG artifacts, leading to a 24% improvement in LPIPS in the CSR case.

**Cleaning the input** Another strategy for tackling the shift between train and test distribution, caused by the bicubic downsampling, is to map the input image $X$ to the bicubic distribution $\hat{Z} = F(X) \sim p_Z$ before applying the SR network, as proposed in [34]. With this strategy, the SR network $S$ is trained using bicubic data, exactly as in the baseline setting discussed previously. During inference, the input image $X$ is super-resolved as $\hat{Y} = S(F(X))$. In fact, in our approach, we already train such a mapping $F$ to ensure cycle consistency in the training of domain correction network $G$ (Section 3.3). Since our training is fully symmetric between the two domains, including the architectures of $F$ and $G$, we use the generator $F$ trained in our framework for a fair comparison.

In case of sensor noise, this version improve the results compared to the baseline method, suggesting that some of the domain shift problem is alleviated. However, our approach further improves the LPIPS distance by 50% in the DSR and 19% in the CSR case. This is partly due to the fact that the SR network acts directly on the input image.
We then perform the direct pixel-wise supervision in the LR domain. Similar to [34], we add another generator network $H$ that maps the super-resolved image $\hat{Y}$ back to the original domain. We observed that this approach leads to stronger GAN hallucinations, as shown in Figure 6. This tendency is also observed in the quantitative results, obtaining significantly worse LPIPS and PSNR in all cases. This demonstrates the importance of direct HR supervision provided by our method.

**Low resolution supervision** Here, we compare our approach with performing supervision in the LR domain. Similar to [34], we add another generator network $H$ that maps the super-resolved image $\hat{Y}$ back to the original domain. We then perform the direct pixel-wise supervision in the LR domain instead, using the same losses $L_1$ and $L_{VGG}$ to ensure $H(\hat{Y}) \approx X$. We observed that this approach leads to stronger GAN hallucinations, as shown in Figure 6. This tendency is also observed in the quantitative results, obtaining significantly worse LPIPS and PSNR in all cases. This demonstrates the importance of direct HR supervision provided by our method.

**Fully supervised** To assess the performance of our approach, we compare with fully supervised training using paired samples, otherwise unavailable to the network. We generate paired data $(\tilde{X}, \tilde{Y})$ using the same strategy employed for evaluation, i.e., by applying the ground-truth degradation. The ESRGAN is then finetuned on this data directly. Note that the ground-truth degradation operation is unknown for all other methods in this comparison. We observe that our approach achieves performance much closer to this upper bound for both DSR and CSR. In particular in the case of JPEG artifacts, where our unsupervised method is only slightly worse than full supervision.

### 4.3. State-of-the-art Comparison on DIV2K

In the following we compare our approach with other state-of-the-art methods: ZSSR [28], EDSR [25], ESRGAN [32]. Therefore we use the original code and trained models. The method ZSSR applies a Zero-Shot learning strategy, where weights are learned for each image individually. The EDSR trains a ResNet-based model without perceptual loss. ESRGAN applies the same SR architecture and perceptual losses as in our approach. Both EDSR and ESRGAN are trained using bicubic supervision.

We also report the results of finetuning the ESRGAN on the same training data employed by our approach. In the case of CSR, where two distinct training sets are available, we further compare with finetuning the ESRGAN on each of those. The method "ESRGAN FT IN" is trained on the set of input images, while "ESRGAN FT OUT" is trained on the set of output images. In all cases we follow the same training procedure as [32], constructing the corresponding LR image using bicubic downsampling.

Evaluated the aforementioned approaches on the DIV2K validation set using the DSR and CSR setting as described in Section 4.1. Results for the DSR and CSR settings are reported in Table 2.

#### Table 2. Comparison to state-of-the-art super-resolution methods on the DIV2K dataset using the DSR and CSR setting. Our approach achieves the best perceptual results in both sensor noise and JPEG artifact case.

<table>
<thead>
<tr>
<th>Method</th>
<th>Sensor Noise</th>
<th>JPEG Artifacts</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
<td>SSIM</td>
</tr>
<tr>
<td>ZSSR</td>
<td>23.65</td>
<td>0.47</td>
</tr>
<tr>
<td>EDSR</td>
<td>23.39</td>
<td>0.44</td>
</tr>
<tr>
<td>ESRGAN</td>
<td>17.17</td>
<td>0.19</td>
</tr>
<tr>
<td>ESRGAN FT</td>
<td>18.08</td>
<td>0.22</td>
</tr>
<tr>
<td>Ours</td>
<td>22.43</td>
<td>0.40</td>
</tr>
<tr>
<td>ZSSR</td>
<td>24.87</td>
<td>0.60</td>
</tr>
<tr>
<td>EDSR</td>
<td>24.46</td>
<td>0.53</td>
</tr>
<tr>
<td>ESRGAN</td>
<td>17.39</td>
<td>0.19</td>
</tr>
<tr>
<td>ESRGAN FT IN</td>
<td>18.35</td>
<td>0.23</td>
</tr>
<tr>
<td>ESRGAN FT OUT</td>
<td>17.35</td>
<td>0.19</td>
</tr>
<tr>
<td>Ours</td>
<td>22.42</td>
<td>0.55</td>
</tr>
</tbody>
</table>
best performance, owing to its zero-shot learning strategy. Our unsupervised approach achieves the best overall perceptual quality, significantly reducing the LPIPS error metric by 58% and 44% for DSR and CSR respectively, in the sensor noise setting.

As shown in Figure 7 the ESRGAN approaches produce strong artifacts in the case of noisy input. This is likely due to the perceptual loss that encourages the network to output high-frequency components in order to provide a sharp image. In contrast, our method do not suffer from such artifacts despite employing a strong perceptual loss. This demonstrates that our SR network has learned significant robustness towards image noise, though our unsupervised training strategy.

**JPEG Compression** In the DSR case, where SR is performed within the same domain, our approach achieves a significantly lower LPIPS distance compared to state-of-the-art. For CSR, where the task is to additionally clean the input from artifacts, our approach has a strong advantage, reducing the LPIPS by more than 0.17. However, as shown in Figure 7, the difference in perceptual quality is not fully captured by the quantitative results. All compared approaches produces highly visible block artifacts, stemming from the JPEG compression of the input image. In contrast, our approach provides visually pleasing output without such artifacts.

### 4.4. Real-World Evaluation on the DPED Dataset

In this section we apply our method to the original images of the DPED iPhone3 dataset [16]. It contains natural images, which include real-world degradations due to poor sensor and lens quality. We train our model using the training split to represent both the input and output distributions. We also finetune the ESRGAN model on the same data, as described in the previous experiment. Since ground-truth images are not available in this real-world setting, we show a diverse set of qualitative examples from the DPED iPhone3 validation set in Figure 8. The artifacts produced by ZSSR, EDSR, ESRGAN and ESRGAN Fine-tuned are of similar nature as in the sensor noise case in the DIV2K setting in Figure 7. Note that these limitations in previous approaches cannot be alleviated by more training data or architectural designs. Instead these issues originate from an oversimplified problem formulation not reflected in most real-world applications. Our approach is able to overcome the limitations of previous methods by learning the input image distribution. Our approach generate high-quality images, with very few artifacts.

### 5. Conclusion

We tackle the problem of real-world super-resolution, where no paired data is available. To avoid the artifacts caused by bicubic downsampling, we learn a network that restores the low resolution image to the real-world image distribution. This allows us to generate realistic training pairs for our super-resolution model. Lastly, we propose a benchmark, based on the DIV2K dataset, for quantitatively evaluating real-world super-resolution approaches. Experiments are performed on our real-world benchmark and the DPED datasets. Compared previous methods, our approach generalizes to natural images, affected by significant sensor noise, compression artifacts and other effects.

**Acknowledgements.** This work was partly supported by ETH General Fund, Amazon through an AWS grant, Nvidia through a GPU grant, and Huawei.

**References**

[1] N. Ahn, B. Kang, and K.-A. Sohn. Fast, accurate, and lightweight super-resolution with cascading residual net-


