Correction Filter for Single Image Super-Resolution: Robustifying Off-the-Shelf Deep Super-Resolvers

Shady Abu Hussein
Tel Aviv University, Israel
shadya@mail.tau.ac.il

Tom Tirer
Tel Aviv University, Israel
tomtirer@mail.tau.ac.il

Raja Giryes
Tel Aviv University, Israel
raja@tauex.tau.ac.il

Abstract

The single image super-resolution task is one of the most examined inverse problems in the past decade. In the recent years, Deep Neural Networks (DNNs) have shown superior performance over alternative methods when the acquisition process uses a fixed known downsampling kernel—typically a bicubic kernel. However, several recent works have shown that in practical scenarios, where the test data mismatch the training data (e.g. when the downsampling kernel is not the bicubic kernel or is not available at training), the leading DNN methods suffer from a huge performance drop. Inspired by the literature on generalized sampling, in this work we propose a method for improving the performance of DNNs that have been trained with a fixed kernel on observations acquired by other kernels. For a known kernel, we design a closed-form correction filter that modifies the low-resolution image to match one which is obtained by another kernel (e.g. bicubic), and thus improves the results of existing pre-trained DNNs. For an unknown kernel, we extend this idea and propose an algorithm for blind estimation of the required correction filter. We show that our approach outperforms other super-resolution methods, which are designed for general downsampling kernels.

1. Introduction

The task of Single Image Super-Resolution (SISR) is one of the most examined inverse problems in the past decade [10, 12, 35, 7]. In this problem, the goal is to reconstruct a latent high-resolution (HR) image from its low-resolution (LR) version, obtained by an acquisition process that includes low-pass filtering and sub-sampling. In the recent years, along with the developments in deep learning, many SISR methods that are based on Deep Neural Networks (DNNs) have been proposed [6, 15, 19, 18, 41, 33, 14].

Typically, the performance of SISR approaches is evaluated on test sets with a fixed known acquisition process, e.g. a bicubic downsampling kernel. This evaluation methodology allows to prepare large training data, which are based on ground truth HR images and their LR counterparts synthetically obtained through the known observation model. DNNs, which have been exhaustively trained on such training data, clearly outperform other alternative algorithms, e.g. methods that are based on hand-crafted prior models such as sparsity or non-local similarity [7, 22, 12, 35, 36].

Recently, several works have shown that in practical scenarios where the test data mismatch the training data, the leading DNN methods suffer from a huge performance drop [37, 26, 29]. Such scenarios include a downsampling kernel which is not the bicubic kernel and is not available at the training phase. A primary example is an unknown kernel that needs to be estimated from the LR image at test time.

Several recent SISR approaches have proposed different strategies for enjoying the advantages of deep learning while mitigating the restriction of DNNs to the fixed kernel assumption made in the training phase. These strategies include: modifying the training phase such that it covers a predefined set of downsampling kernels [38, 13]; using DNNs to capture only a natural-image prior which is decoupled from the SISR task [37, 4]; or completely avoid any offline training and instead train a CNN super-resolver from scratch at test time [30, 26].

Contribution. In this work we take a different strategy, inspired by the generalized sampling literature [8, 31], for handling LR images obtained by arbitrary downsampling kernels. Instead of completely ignoring the state-of-the-art DNNs that have been trained for the bicubic model, as done by other prior works, we propose a method that transforms the LR image to match one which is obtained by the bicubic kernel. The modified LR can then be inserted into existing leading super-resolvers, such as DBPN [14], RCAN [41], and proSR [33], thus, improving their performance significantly on kernels they have not been trained on. The proposed transformation is performed using a correction filter, which has a closed-form expression when the true (non-bicubic) kernel is given.

In the “blind” setting, where the kernel is unknown, we...
extend our approach and propose an algorithm that estimates the required correction. The proposed approach outperforms other super-resolution methods in various practical scenarios. See example results in Figure 1 in the case of a known kernel.

2. Related Work

In the past five years many works have employed DNNs for the SISR task, showing a great advance in performance with respect to the reconstruction error [6, 14, 15, 19, 33, 41] and the perceptual quality [3, 18, 24, 32]. However, one main disadvantage of DNN-based SISR methods is their sensitivity to the LR image formation model. A network performance tends to drop significantly if it has been trained for one acquisition model and then been tested on another [26, 29, 37].

Recently, different SISR strategies has been proposed with the goal of enjoying the advantages of deep learning while mitigating the restriction of DNNs to the fixed kernel assumption made in the training phase. One strategy is to train a CNN super-resolver that gets as inputs both the LR image and the degradation model and assumes that the downsampling kernels belong to a certain set of Gaussian filters [13, 38]. Another approach builds on the structural prior of CNNs, which promotes signals with spatially recurring patterns (e.g. natural images) and thus allows to train a super-resolver CNN from scratch at test time [26, 30]. Another line of work recovers the latent HR image by minimizing a cost function, composed of fidelity and prior terms, where only the prior term is handled by a pre-trained deep denoiser or GAN [37, 28, 4, 39]. Recently, the two last approaches have been incorporated [29, 1]. In all these meth-
in the blind setting (where the kernel is unknown) it is still possible to apply these methods after an initial kernel estimation phase.

Our approach is inspired by the literature on generalized sampling [5, 31, 8], which generalizes the classical WhittakerNyquistKotelnikovShannon sampling theorem [34, 21, 17, 25], which considers signals that are band-limited in the frequency domain and sinc interpolations. The generalized theory provides a framework and conditions under which a signal that is sampled by a certain basis can be reconstructed by a different basis. In this framework, the sampled signal is reconstructed using a linear operator that can be decoupled into two phases, the first applies a digital correction filter and the second include processing with a reconstruction kernel. The role of the correction filter is to transform the sampling coefficients, associated with the sampling kernel, to coefficients which fit the reconstruction kernel.

Several works have used the correction filter approach for image processing [9, 11, 23]. These works typically propose linear interpolation methods, i.e. the correction filter is followed by a linear reconstruction operation, and do not use a strong natural-image prior. As a result, the recovery of fine details is lacking.

In this work, we plan to use (very) non-linear reconstruction methods, namely—DNNs, whose training is difficult, computationally expensive, storage demanding, and cannot be done when the observation model is not known in advance. To tackle these difficulties, we revive the correction filter approach and show how it can be used with deep super-resolvers which have been already trained.

The required correction filter depends on the kernel which is used for sampling. Therefore, in the blind setting, it needs to be estimated from the LR image. To this end, we propose an iterative optimization algorithm for estimating the correction filter. In general, only a few works have considered the blind SISR setting and developed kernel estimation methods [27, 20, 13, 2].

Finally, we would like to highlight major differences between this paper and the work in [13], whose "kernel correction" approach may be misunderstood as our "correction filter". In [13], three different DNNs (super-resolver, kernel estimator, and kernel corrector) are offline trained under the assumption that the downsampling kernel belongs to a certain family of Gaussian filters (similarly to [38]), and the CNN super-resolver gets the estimated kernel as an input. So, the first major difference is that contrary to our approach, no pre-trained existing DNN methods (other than SRMD [38]) can be used in [13]. Secondly, their approach is restricted by the offline training assumptions to very certain type of downsampling kernels, contrary to our approach. Thirdly, the concepts of these works are very different: The (iterative) correction in [13] modifies the estimated downsampling kernel, while our correction filter modifies the LR image.

### 3. The Proposed Method

The single image super resolution (SISR) acquisition model, can be formulated as

\[
y = (x \ast k) \downarrow_{\alpha},
\]

where \( x \in \mathbb{R}^n \) represents the latent HR image, \( y \in \mathbb{R}^m \) represents the observed LR image, \( k \in \mathbb{R}^d \) is the (anti-aliasing) blur kernel, \( \ast \) denotes the convolution operator, and \( \downarrow_{\alpha} \) denotes sub-sampling operator with stride of \( \alpha \). Under the common fashion of dropping the edges of \( x \ast k \), such that it is in \( \mathbb{R}^n \), we have that \( m = \lceil n/\alpha \rceil \).

Note that Equation (1) can be written in a more elegant way as

\[
y = S^*x,
\]

where \( S^*: \mathbb{R}^n \rightarrow \mathbb{R}^m \) is a linear operator that encapsulates the entire sampling operation, i.e. \( S^* \) is a composition of blurring followed by sub-sampling. The sampling operator \( S^* \) is presented in Figure 2(a).

Most SISR deep learning methods, e.g. [6, 15, 19, 18, 41, 33, 14], assume that the observations are obtained using the bicubic kernel. Let us denote by \( R^* \) the associated...
sampling operator (essentially, $R^*$ coincides with the previously defined $S^*$ if $k$ is the bicubic kernel $k_{bicub}$). The core idea of our approach is to modify the observations $y = S^*x$, obtained for an arbitrary downsampling kernel $k$, such that they mimic the (unknown) "ideal observations" $y_{bicub} = R^*x$, which can be fed into pre-trained DNN models.

In what follows, we present a method to (approximately) achieve this goal using the correction filter tool, adopted from the generalized sampling literature. First, we consider the non-blind setting, where the downsampling kernel is known, and thus $S^*$ is known. In this case, we obtain a closed-form expression for the required correction filter, which depends on $k$ (and on $k_{bicub}$). Later, we extend the approach to the blind setting, where $k$ is unknown. In this case, we propose a technique for estimating the correction filter from the LR image $y$.

### 3.1. The non-blind setting

In the non-blind setting, both the downsampling kernel $k$ and the target kernel $k_{bicub}$ are known. Therefore, the sampling operators $S^*$ and $R^*$ are known as well. Using common notations from generalized sampling literature \cite{8}, let us denote by $S$ and $R$ the adjoint operators of $S^*$ and $R^*$, respectively. The operator $R : \mathbb{R}^m \to \mathbb{R}^n$ can be interpreted as a reconstruction operator, that restores a signal in $\mathbb{R}^n$ from $m$ samples, associated with the sampling operator $R^*$. In the context of our work, when $R$ is applied on a vector it pads it with $n - m$ zeros ($\alpha$ zeros between each two entries) and convolves it with a flipped version of $k_{bicub}$. The reconstruction operator $R^*$ is presented in Figure 2(c). A similar interpretation goes for $S : \mathbb{R}^m \to \mathbb{R}^n$ with the kernel $k$.

The key goal of generalized sampling is to identify signal models and sampling systems (i.e. sampling and reconstruction operators) that allow for perfect recovery. Therefore, to proceed, let us make the following assumption.

**Assumption 1.** The signal $x$ sampled by $R^*$, can be perfectly recovered by $R$, i.e.

$$x = R R^* x. \quad (3)$$

Even though Assumption 1 does not hold for natural images, it is motivated by the fact that downsampling with a bicubic kernel produces a naturally looking LR image (typically better than the LR images obtained using other kernels), and we have many DNN methods that can handle observations of the form $R^*x$ quite well.

Suppose that we have been given the observations $y_{bicub} = R^*x$ and that Assumption 1 holds, then the estimator $\hat{x} = R y_{bicub}$ would yield perfect reconstruction. However, since we are given different observations, $y = S^*x$, let us propose a different estimator $\hat{x} = R H y$, where $H : \mathbb{R}^m \to \mathbb{R}^m$ is a correction operation. This recovery procedure is presented in Figures 2(b)+2(c). The following theorem presents a condition and a formula for $H$ under which perfect recovery is possible.

**Theorem 2.** Let $y = S^*x$, $\hat{x} = R H y$, and assume that Assumption 1 hold. Then, if

$$\text{null}(S^*) \cap \text{range}(R) = \{0\}, \quad (4)$$

we have that $\hat{x} = x$ for

$$H = (S^* R)^{-1} : \mathbb{R}^m \to \mathbb{R}^m. \quad (5)$$

**Proof.** Note that

$$\hat{x} = R H y = R H S^* x = R H S^* R R^* x, \quad (6)$$

where the last equality follows from Assumption 1. Next, (4) implies that the operator $(S^* R)$ is invertible. Thus, setting $H$ according to (5) is possible, and we get

$$\hat{x} = R (S^* R)^{-1} S^* R R^* x = R R^* x = x, \quad (7)$$

where the last equality follows from Assumption 1.

Theorem 2 is presented in operator notations to simplify the derivation. In the context of SISR (i.e. with the previous definitions of $S$ and $R$), the operator $H = (S^* R)^{-1}$ can be applied simply as a convolution with a correction filter $h$, whose Discrete Fourier Transform (DFT) is given by

$$h = \text{IDFT} \left\{ \frac{1}{\text{DFT} \{ (k \ast k_{bicub}) \downarrow \alpha \}^*} \right\}, \quad (8)$$

where $\text{DFT}(\cdot)$ and $\text{IDFT}(\cdot)$ denote the Discrete Fourier Transform and its inverse respectively. To ensure numerical stability, we slightly modify (9), and compute $h$ using

$$h = \text{IDFT} \left\{ \frac{\text{DFT} \{ (k \ast k_{bicub}) \downarrow \alpha \}^*}{\text{DFT} \{ (k \ast k_{bicub}) \downarrow \alpha \}^2 + \epsilon} \right\}, \quad (9)$$

where $\epsilon$ is a small regularization parameter (in all of our experiments we set it to $10^{-4}$).

In practice, instead of using the weak estimator $\hat{x} = R H y$ that does not use any natural-image prior, we propose to recover the HR image by

$$\hat{x} = f(h \ast y), \quad (10)$$

where $f(\cdot)$ is a DNN super-resolver that has been trained under the assumption of bicubic kernel (i.e. we replace the
Table 1: Non-blind super-resolution comparison on Set14. Each cell displays PSNR [dB] (left), SSIM (middle) and perceptual distance [40] (right).

| scale | Gaussian std = 1.5/√2 | Gaussian std = 2.5/√2 | Box of width = 4 |
|-------|------------------------|------------------------|-----------------|
| DBPN  | 30.078 / 0.85 / 0.109  | 26.366 / 0.734 / 0.219 | 28.444 / 0.803 / 0.153 |
| DBPN + Correction | 33.686 / 0.904 / 0.041 | 31.473 / 0.881 / 0.054 | 29.147 / 0.819 / 0.141 |
| proSR | 30.073 / 0.849 / 0.108 | 26.371 / 0.734 / 0.218 | 28.459 / 0.803 / 0.151 |
| proSR + Correction | 33.626 / 0.903 / 0.042 | 31.582 / 0.883 / 0.055 | 29.165 / 0.820 / 0.140 |
| RCAN  | 30.118 / 0.851 / 0.107 | 26.389 / 0.736 / 0.219 | 28.469 / 0.804 / 0.152 |
| RCAN + Correction | 33.654 / 0.903 / 0.041 | 31.272 / 0.878 / 0.053 | 29.140 / 0.820 / 0.141 |
| ZSSR  | 28.107 / 0.829 / 0.066 | 27.954 / 0.806 / 0.100 | 28.506 / 0.802 / 0.135 |
| SRMD  | 2.64 / 0.878 / 0.078  | 29.923 / 0.812 / 0.133 | 25.944 / 0.757 / 0.112 |

| scale | Gaussian std = 3.5/√2 | Gaussian std = 4.5/√2 | Box of width = 8 |
|-------|------------------------|------------------------|-----------------|
| DBPN  | 25.067 / 0.685 / 0.254 | 23.890 / 0.645 / 0.293 | 24.636 / 0.667 / 0.265 |
| DBPN + Correction | 28.385 / 0.773 / 0.141 | 27.235 / 0.760 / 0.155 | 25.181 / 0.666 / 0.284 |
| proSR | 25.033 / 0.683 / 0.254 | 23.882 / 0.645 / 0.292 | 24.685 / 0.667 / 0.265 |
| proSR + Correction | 28.369 / 0.772 / 0.143 | 27.445 / 0.757 / 0.158 | 25.205 / 0.670 / 0.284 |
| RCAN  | 25.077 / 0.685 / 0.257 | 23.904 / 0.646 / 0.296 | 24.694 / 0.668 / 0.268 |
| RCAN + Correction | 28.168 / 0.769 / 0.144 | 27.076 / 0.757 / 0.155 | 25.236 / 0.671 / 0.289 |
| ZSSR  | 25.642 / 0.701 / 0.181 | 25.361 / 0.683 / 0.198 | 24.549 / 0.653 / 0.203 |
| SRMD  | 26.877 / 0.718 / 0.200 | 26.350 / 0.674 / 0.236 | 19.704 / 0.525 / 0.275 |

Algorithm 1: Correction filter estimation

**Input:** y, \( k_{bicub}, N_{iter} \)

**Output:** h an estimate for h.

**Params.:** \( k_0 = \) Gaussian kernel with \( \sigma = 1, i = 0 \), \( \lambda = 0.01 \).

\[ R = \text{DFT}\{k_{bicub}\}; \]

while \( i < N_{iter} \) do

\[ i = i + 1; \]

\[ h_i = \arg \min_y \| y - (k_{i-1} * k_{bicub} * ((h_i + y) \uparrow_\alpha)) \downarrow_\alpha \|_1; \]

\[ h = \text{DFT}\{h_i\} \uparrow_\alpha; \]

\[ k_i = \arg \min_k \| 1 - H \cdot R \cdot \text{DFT}\{k\} \|_1 + \lambda \| k \|_1; \]

end

\[ h = h_i; \]

All images were cropped to a size of 512x512 pixels. The naive reconstruction operator in Figure 2(c) was replaced with a DNN.

For the experiments in this paper we use DBPN [14], RCAN [41], and proSR [33], but in general any other method with state-of-the-art performance (for bicubic kernel) is expected to give good results.

Note that the theoretical motivation for our strategy requires that the condition in (4) holds. This condition can be inspected by comparing the bandwidth of the kernels \( k \) and \( k_{bicub} \) in the frequency domain. As \( k \) is a low-pass filter (and so is \( k_{bicub} \)), the condition requires that the passband of \( k_{bicub} \) is contained in the passband of \( k \). Yet, as shown in the experiments section, our approach yields a significant improvement even when the passband of \( k \) is moderately smaller than the passband of \( k_{bicub} \).

Furthermore, we observe that even for very blurry LR images performance improvement can be obtained by increasing the regularization parameter in (9).

**3.2. The blind setting**

In the blind setting, the downsampling kernel \( k \) is unknown. Therefore, we cannot compute the correction filter \( h \) using (9), and extending our approach to this setting requires estimating \( h \) from the LR image \( y \). To this end, we propose an iterative optimization algorithm that alternates between obtaining \( h \) that minimizes \( \| y - S^* H y \|_1 \) (note that \( H \) depends on \( h \)), and obtaining \( k \) that minimizes \( \| I - HS^* R \|_1 + \lambda \| k \|_1 \) (note that \( S^* \) depends on \( k \)). A detailed description of the algorithm is given in Algorithm 1. The first inner minimization is performed using Adam [16] with learning rate of \( 10^{-5} \) for 50 iterations. The second inner minimization is performed using SGD with learning rate of 1.0 for 10 iterations at the beginning and then the iterations number is increased by 10 after each outer iteration (limited to 500). We also use \( N_{iter} = 30 \) for scale factor of 2, and \( N_{iter} = 50 \) for scale factor of 4.

After an estimator of \( h \) is obtained using Algorithm 1, the HR image can be reconstructed by (10), similarly to the non-blind setting.
Table 2: Non-blind super-resolution comparison on BSD100. Each cell displays PSNR [dB] (left), SSIM (middle) and perceptual distance [40] (right).

| Method              | Scale | Gaussian std = 1.5/√2 | Gaussian std = 2.5/√2 |
|---------------------|-------|------------------------|------------------------|
| DBPN                | 2     | 26.371 / 0.711 / 0.216 | 30.702 / 0.866 / 0.069 |
| DBPN + correction   | 2     | 26.381 / 0.711 / 0.216 | 30.773 / 0.867 / 0.069 |
| proSR               | 2     | 26.381 / 0.711 / 0.216 | 30.773 / 0.867 / 0.069 |
| proSR + correction  | 2     | 27.418 / 0.738 / 0.156 | 30.773 / 0.867 / 0.069 |
| RCAN                | 2     | 26.371 / 0.711 / 0.216 | 30.773 / 0.867 / 0.069 |
| RCAN + correction   | 2     | 26.381 / 0.711 / 0.216 | 30.773 / 0.867 / 0.069 |
| SRMD                | 2     | 26.371 / 0.711 / 0.216 | 30.773 / 0.867 / 0.069 |

Table 3: Blind super-resolution comparison on Set14. Each cell displays PSNR [dB] (left), SSIM (middle) and perceptual distance [40] (right).

| Method              | Scale | Gaussian std = 3.5/√2 | Gaussian std = 4.5/√2 | Box of width = 4 |
|---------------------|-------|------------------------|------------------------|-----------------|
| DBPN                | 4     | 25.281 / 0.663 / 0.248 | 27.804 / 0.774 / 0.203 | 29.129           |
| DBPN + correction   | 4     | 27.513 / 0.739 / 0.155 | 27.124 / 0.732 / 0.164 | 29.129           |
| proSR               | 4     | 27.513 / 0.739 / 0.155 | 27.124 / 0.732 / 0.164 | 29.129           |
| proSR + correction  | 4     | 27.513 / 0.739 / 0.155 | 27.124 / 0.732 / 0.164 | 29.129           |
| RCAN                | 4     | 27.513 / 0.739 / 0.155 | 27.124 / 0.732 / 0.164 | 29.129           |
| RCAN + correction   | 4     | 27.513 / 0.739 / 0.155 | 27.124 / 0.732 / 0.164 | 29.129           |
| ZSSR                | 4     | 25.115 / 0.651 / 0.201 | 24.348 / 0.625 / 0.246 | 29.129           |
| SRMD                | 4     | 25.735 / 0.704 / 0.187 | 26.432 / 0.707 / 0.189 | 29.129           |

4. Experiments

In this section we examine the performance and improvement due to our correction filter approach in the non-blind and blind settings, using three different off-the-shelf DNN super-resolvers that serve as f(·) in (10): DBPN [14], RCAN [41], and proSR [33]. We compare our approach to other methods that receive the downsampling kernel \( h \). The results are presented in Tables 1 and 2 for the test-sets Set14 and BSD100, respectively. Figures 1 and 3, as well as Appendix A, present several visual results. It can be seen that the proposed filter correction approach significantly improves the results of DBPN, RCAN, and proSR.

4.1. The non-blind setting

In this section, we assume that the downsampling kernel \( k \) is known. Therefore, the correction filter \( h \) can be computed directly using (9). We examine scenarios with scale factors of 2 and 4. For scale factor of 2, we use Gaussian kernels with standard deviation \( \sigma = 1.5/\sqrt{2} \) and \( \sigma = 2.5/\sqrt{2} \), and box kernel of size \( 4 \times 4 \). For scale factor of 4, we use Gaussian kernels with standard deviation \( \sigma = 3.5/\sqrt{2} \) and \( \sigma = 4.5/\sqrt{2} \), and box kernel of size \( 8 \times 8 \).

The results are presented in Tables 1 and 2 for the test-sets Set14 and BSD100, respectively. Figures 1 and 3, as well as Appendix A, present several visual results. It can be seen that the proposed filter correction approach significantly improves the results of DBPN, RCAN, and proSR. However, since its results are extremely inferior to the other strategies (e.g. about 10 dB lower PSNR) they are deferred to Appendix A. All the experiments are performed with the official code of each method. Unfortunately, such code has not been available for [13].
Figure 3: Non-blind super-resolution of image 189080 from BSD100, for scale factor of 2 and Gaussian downsampling kernel with standard deviation $2.5/\sqrt{2}$. From left to right and top to bottom: original image, cropped zoom of the original image, bicubic upsampling, SRMD [38], ZSSR [26], proSR [33], RCAN [41], DBPN [14], proSR with correction, RCAN with correction. which have been trained for the (incorrect) bicubic kernel. Moreover, note that the filter-corrected applications of DBPN, RCAN, and proSR, also outperform SRMD and ZSSR, while the plain applications of DBPN, RCAN, and proSR are inferior to SRMD.

Inference run-time. Computing the correction filter requires a negligible amount of time, so it does not change the run-time of an off-the-shelf DNN. Using NVIDIA RTX 2080ti GPU, the per image run-time of all the methods except ZSSR is smaller than 1 second (because no training is done in the test phase), while ZSSR requires approximately 2 minutes per image.

The spectrums of sampling kernels. In Figure 4 we present the frequency spectrums of Gaussian kernels that are used in this work and two bicubic kernels, one for scale factor of 2 and one for scale factor of 4. In order to ease the presentation, the curves are plotted for the one-dimensional version of these kernels. Our experiments show that for super-resolution factor of 2 the proposed filter correction approach yields very good results for Gaussian sampling kernels of $1.5/\sqrt{2}$ and $2.5/\sqrt{2}$ (see Tables 1 and 2), but not for $3.5/\sqrt{2}$ (not shown). This behavior correlates with the recovery guarantees in Theorem 2. The theory essentially requires that the passband of the bicubic kernel (associated with relevant scale factor) is contained in the passband of the downsampling kernels. Indeed, it can be seen in Figure
that the passband of the bicubic kernel for scale factor 2 is contained in the passband of the Gaussian with $1.5/\sqrt{2}$, but extremely larger than the passband of the Gaussian with $3.5/\sqrt{2}$. On the other hand, for super-resolution factor of 4 we do get good results for Gaussian sampling kernel of $3.5/\sqrt{2}$ (see Tables 1 and 2). This is aligned with the fact that the bicubic kernel for scale factor 4 is much narrower in frequency domain than the bicubic kernel for scale factor 2, as shown in Figure 4.

We note that the deficiency of the correction filter approach for the setting with Gaussian kernel of $\sigma = 3.5/\sqrt{2}$ and scale factor of 2 can be resolved by increasing the regularization parameter $\epsilon$ used in (9), which helps compensating the null space in $S^* R$ and stabilizes its inverse. For example, plain application of DBPN on Set14 yields PSNR of 24.71 dB, and applying it after correction filter with $\epsilon = 0$ yields PSNR of 10.34 dB. However, applying DBPN after correction filter with $\epsilon = 10$ yields PSNR of 28.67 dB.

4.2. The blind setting

In this section, we turn to test non-blind super-resolution, where the downsampling kernel $k$ is unknown. Therefore, to apply our approach we first estimate the correction filter using Algorithm 1, and then use this estimation to restore the HR image by (10). In this setting we compare our method to kernelGAN [2], which estimates the downsampling kernel using adversarial training (in test-time) and then uses ZSSR to restore the HR image.

The results for Set14 are presented in Table 3 and visual examples are shown in Figure 5 and in Appendix B. It can be seen that the proposed filter correction approach still improves the results of DBPN, RCAN, and proSR, compared to their plain applications (which are presented in Tables 1). It also outperforms kernelGAN (despite our approach being a much simpler) in terms of reconstruction error and gains comparable results in terms of perceptual distance. Our proposed strategy has an advantage over kernelGAN also in term of run-time. While kernelGAN requires about 6.5 minutes per image, estimating the correction filter requires 2 minutes, which is followed by off-the-shelf DNNs that require less than 1 second.

5. Conclusion

The SISR task has gained a lot from the developments in deep learning in the recent years. Yet, the leading DNN methods suffer from a huge performance drop when they are tested on images that do not fit the acquisition process assumption used in their training phase—which is, typically, that the downsampling kernel is bicubic. In this work, we addressed this issue by a signal processing approach: computing a correction filter that modifies the low-resolution observations such that they mimic observations that are obtained with a bicubic kernel. (Notice that our focus in this work on the bicubic kernel is for the sake of simplicity of the presentation and due to its popularity. Yet, it is possible to use our developed tools also for other reconstruction kernels). The modified LR is then fed into existing state-of-the-art DNNs that are trained only under the assumption of bicubic kernel. Various experiments have shown that the proposed approach significantly improves the performance of the pre-trained DNNs and outperforms other (much more sophisticated) methods that are specifically designed to be robust to different kernels.

Acknowledgments

This work is partially supported by the NSF-BSF grant (No. 2017729) and the European research council (ERC StG 757497 PI Giryes).

References

[1] Shady Abu Hussein, Tom Tirer, and Raja Giryes. Image-adaptive GAN based reconstruction. arXiv preprint arXiv:1906.05284, 2019.
[2] Sefi Bell-Kligler, Assaf Shocher, and Michal Irani. Blind super-resolution kernel estimation using an internal-GAN. NeurIPS, 2019.
[3] Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. IEEE Conference on Computer Vision and Pattern Recognition, Jun. 2018.
[4] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G Dimakis. Compressed sensing using generative models. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pages 537–546. JMLR. org, 2017.
Figure 5: Blind super-resolution of baboon image from Set14, for scale factor of 2 and Gaussian downsampling kernel with standard deviation $1.5/\sqrt{2}$. From left to right and top to bottom: original image, cropped zoom of the original image, bicubic upsampling, kernelGAN [2], proSR[33], RCAN[41], DBPN[14], proSR with estimated correction, RCAN with estimated correction and DBPN with estimated correction.

[5] Paul Leo Butzer. A survey of the Whittaker-Shannon sampling theorem and some of its extensions. J. Math. Res. Expo., 1983.

[6] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In European conference on computer vision, pages 184–199. Springer, 2014.

[7] Weisheng Dong, Lei Zhang, Guangming Shi, and Xin Li. Nonlocally centralized sparse representation for image restoration. IEEE transactions on Image Processing, 22(4):1620–1630, 2012.

[8] Yonina C Eldar. Sampling theory: Beyond bandlimited systems. Cambridge University Press, 2015.

[9] Yonina C Eldar and Tomer Michaeli. Beyond bandlimited sampling. IEEE signal processing magazine, 26(3):48–68, 2009.

[10] William T Freeman, Thouis R Jones, and Egon C Pasztor. Example-based super-resolution. IEEE Computer graphics...
and Applications, (2):56–65, 2002. 1

[11] CA Glasbey. Optimal linear interpolation of images with known point spread function. In Proceedings of the Scandinavian Conference on Image Analysis., pages 161–168, 2001. 3

[12] Daniel Glasner, Shai Bagon, and Michal Irani. Super-resolution from a single image. In 2009 IEEE 12th international conference on computer vision, pages 349–356. IEEE. 1

[13] Jinjin Gu, Hannan Lu, Wangmeng Zuo, and Chao Dong. Blind super-resolution with iterative kernel correction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1604–1613, 2019. 1, 2, 3, 6

[14] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Deep back-projection networks for super-resolution. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1664–1673, 2018. 1, 2, 3, 5, 6, 7, 9, 11

[15] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1646–1654, 2016. 1, 2, 3

[16] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. 5

[17] Vladimir Aleksandrovich Kotelnikov. On the transmission capacity of the ether and of cables in electrical communications. 1933. 3

[18] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4681–4690, 2017. 1, 2, 3

[19] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pages 136–144, 2017. 1, 2, 3

[20] Tomer Michaeli and Michal Irani. Nonparametric blind super-resolution. In Proceedings of the IEEE International Conference on Computer Vision, pages 945–952, 2013. 3

[21] Harry Nyquist. Certain topics in telegraph transmission theory. Transactions of the American Institute of Electrical Engineers, 47(2):617–644, 1928. 3

[22] M. Protter, M. Elad, H. Takeda, and P. Milanfar. Generalizing the non-local-means to super-resolution reconstruction. IEEE Transactions on Image Processing, 18(1):36–51, 2009. 1

[23] Sathish Ramani, Dimitri Van De Ville, and Michael Unser. Non-ideal sampling and adapted reconstruction using the stochastic matern model. In 2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings, volume 2, pages II–II. IEEE, 2006. 3

[24] Mehdi SM Sajjadi, Bernhard Scholkopf, and Michael Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. In Proceedings of the IEEE International Conference on Computer Vision, pages 4491–4500, 2017. 2

[25] Claude Elwood Shannon. Communication in the presence of noise. Proceedings of the IRE, 37(1):10–21, 1949. 3

[26] Assaf Shocher, Nadav Cohen, and Michal Irani. zero-shot super-resolution using deep internal learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3118–3126, 2018. 1, 2, 6, 7, 11

[27] Filip Sroubek, Gabriel Cristóbal, and Jan Flusser. A unified approach to super-resolution and multichannel blind deconvolution. IEEE Transactions on Image Processing, 16(9):2322–2332, 2007. 3

[28] Tom Tirer and Raja Giryes. Image restoration by iterative denoising and backward projections. IEEE Transactions on Image Processing, 28(3):1220–1234, 2018. 2

[29] Tom Tirer and Raja Giryes. Super-resolution via image-adapted denoising CNNs: Incorporating external and internal learning. IEEE Signal Processing Letters, 2019. 1, 2

[30] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 9446–9454, 2018. 1, 2

[31] Michael Unser. Sampling50 years after shannon. Proceedings of the IEEE, 88(ARTICLE):569–587, 2000. 1, 3

[32] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In Proceedings of the European Conference on Computer Vision (ECCV), pages 0–0, 2018. 2

[33] Yifan Wang, Federico Perazzi, Brian McWilliams, Alexander Sorkine-Hornung, Olga Sorkine-Hornung, and Christopher Schroers. A fully progressive approach to single-image super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 864–873, 2018. 1, 2, 3, 5, 6, 7, 9, 11

[34] Edmund Taylor Whittaker. On the functions which are represented by the expansions of the interpolation-theory. Proceedings of the Royal Society of Edinburgh, 35:181–194, 1915. 3

[35] Jianhao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. IEEE transactions on image processing, 19(11):2861–2873, 2010. 1

[36] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In Proceedings of the 7th International Conference on Curves and Surfaces, pages 711–730. Springer, 2010. 1

[37] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep CNN denoiser prior for image restoration. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3929–3938, 2017. 1, 2

[38] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Learning a single convolutional super-resolution network for multiple degradations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3262–3271, 2018. 1, 2, 3, 6, 7, 11
A. Additional results for the non-blind setting

Additional visual results of the different methods in the non-blind setting (which are not included in the paper due to space limitation) are presented in Figures 7 and 8.

We also present in Figure 6, the results obtained by the DPSR method [39] (using its official implementation code). It can be seen that DPSR results have many artifacts. In fact, DPSR average PSNR as produced by its official code is lower by more than 10 dB than the other examined methods (SRMD [38], ZSSR [26], DBPN [14], proSR [33], RCAN [41], and our proposed approach). Therefore, it is not displayed in the tables in the paper.

B. Additional results for the blind setting

Additional visual results of the different methods in the blind setting (which are not included in the paper due to space limitation) are presented in Figures 9 and 10.
Figure 6: DPSR [39] results for non-blind super-resolution of baboon and monarch images from Set14, for scale factor of 2 and Gaussian downsampling kernels. Left is with filter of std $1.5/\sqrt{2}$, and right is with filter of std $2.5/\sqrt{2}$.

Figure 7: Non-blind super-resolution of image 19021 from BSD100, for scale factor 4 and Gaussian downsampling kernel with std $4.5/\sqrt{2}$. From left to right and top to bottom: original image, cropped zoom of the original image, bicubic upsampling, SRMD, ZSSR, proSR, RCAN, DBPN, proSR with correction, RCAN with correction, DBPN with correction.
Figure 8: Non-blind super-resolution of image 148026 from BSD100, for scale factor 4 and Gaussian downsampling kernel with std $3.5/\sqrt{2}$. From left to right and top to bottom: original image, cropped zoom of the original image, bicubic upsampling, SRMD, ZSSR, proSR, RCAN, DBPN, proSR with correction, RCAN with correction, DBPN with correction.
Figure 9: Blind super-resolution of comic from Set14, for scale factor 2 and box downsampling kernel of width 4. From left to right and top to bottom: original image, cropped zoom of the original image, bicubic upsampling, kernelGAN, proSR, RCAN, DBPN, proSR with correction, RCAN with correction, DBPN with correction.
Figure 10: Blind super-resolution of man image from Set14, for scale factor 4 and Gaussian downsampling kernel of width $3.5/\sqrt{2}$. From left to right and top to bottom: original image, cropped zoom of the original image, bicubic upsampling, kernelGAN, proSR, RCAN, DBPN, proSR with correction, RCAN with correction, DBPN with correction.