Conditioned Regression Models for Non-Blind Single Image Super-Resolution

Gernot Riegler Samuel Schulter Matthias Rüther Horst Bischof
Institute for Computer Graphics and Vision, Graz University of Technology
{riegler,schulter,ruether,bischof}@icg.tugraz.at

Abstract

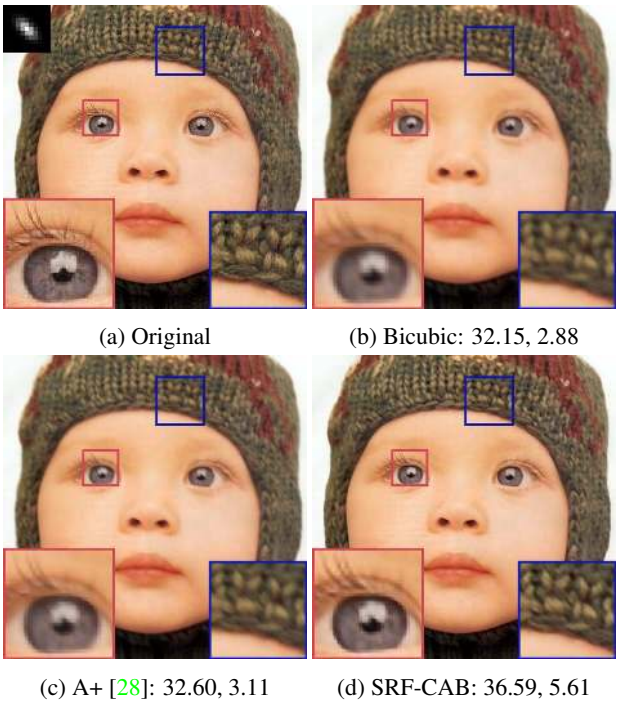
Single image super-resolution is an important task in the field of computer vision and finds many practical applications. Current state-of-the-art methods typically rely on machine learning algorithms to infer a mapping from low- to high-resolution images. These methods use a single fixed blur kernel during training and, consequently, assume the exact same kernel underlying the image formation process for all *test images. However, this setting is not realistic for practical applications, because the blur is typically different for each test image. In this paper, we loosen this restrictive constraint and propose conditioned regression models (including convolutional neural networks and random forests) that can effectively exploit the additional kernel information during both, training and inference. This allows for training a single model, while previous methods need to be re-trained for every blur kernel individually to achieve good results, which we demonstrate in our evaluations. We also empirically show that the proposed conditioned regression models (i) can effectively handle scenarios where the blur kernel is different for each image and (ii) outperform related approaches trained for only a single kernel.*

1. Introduction

Single image super-resolution (SISR) is a classical problem in computer vision and has attracted a lot of attention in recent years. The task is to recover a high-resolution image h given the corresponding low-resolution image l. These two quantities are related as

$$l = \downarrow(k * h) \,, \tag{1}$$

where k is a blur kernel, $*$ denotes the convolution operator, and $\downarrow$ the down-sampling operator. The problem is inherently ill-posed as a single low-resolution image can map to several high-resolution images. One way to tackle this problem is by assuming some prior on the high-resolution image h, *e.g.*, smoothness [10, 30], or by learning a mapping from low- to high-resolution in a data-driven manner. The latter approach is typically used in recent state-of-the-art methods, where dictionary learning approaches [28], random forests [24], or neural networks [6] are employed.



(a) Original (b) Bicubic: 32.15, 2.88

(c) A+ [28]: 32.60, 3.11 (d) SRF-CAB: 36.59, 5.61

Figure 1: Low-resolution images can underlie different blur kernels in their formation process and should be treated accordingly [18]. This figure illustrates the effect of a wrong blur kernel assumption on the super-resolution quality (the original blur kernel is depicted in the top left corner of (a)). Not only standard bicubic upscaling but also state-of-the-art methods like [28] trained on a wrong kernel yield clearly inferior results (PSNR and IFC [26] are given) compared to our conditioned regression model (SRF-CAB).

SISR can be divided into two different sub-problems. First, *blind* super-resolution algorithms consider k and h unknown and try to recover both quantities from the low-resolution image l. Second, *non-blind* super-resolution algorithms assume k to be known and only recover h from

both k and l. There exist only a few works on blind super-resolution [12, 18, 27, 31] and most of them focus on recovering the blur kernel k and use existing non-blind methods for the final super-resolution step. Such a two-step strategy is also advocated for general deconvolution because a joint recovery of k and h is often hard and yields suboptimal results for the finally desired high-resolution image h [17].

Thus, a non-blind super-resolution algorithm should be able to effectively handle low-resolution images underlying different blur kernels. There exist SISR methods that can incorporate any (also previously unseen) blur kernels during inference. Most of these methods rely on the re-occurrence of patches within the image and across scales, *e.g.*, [11, 13], and do not require any training phase. These approaches work remarkably well on repetitive structures but do not achieve state-of-the-art results on natural images [13]. The currently best performing methods in non-blind SISR typically rely on machine learning techniques to directly learn a mapping from low- to high-resolution images from a large corpus of training data [6, 24, 28, 29]. One problem though is that all these methods not only assume the blur kernel k to be known, but also to be equal for all training and test images. Thus, these methods are designed for a single blur kernel only, *e.g.*, bicubic or an isotropic Gaussian with a fixed kernel width. Adapting these methods for different blur kernels almost always requires a separate training phase of the model. This can take a long time, ranging from several minutes [24] to even days [6]. To recap, effectively upscaling a single image with a given, but previously unseen blur kernel would require a separate training phase, which is obviously impractical for real-world applications.

In this work, we analyze recent learning-based methods [6, 24, 28, 29] for the task of non-blind SISR without the assumption of a fixed blur kernel for each image during inference. We present adaptations to these machine learning based methods, such that they can be conditioned on a given blur kernel k during both, training and inference. In particular, we focus on a convolutional neural network and a random forest formulation and show that both can effectively incorporate the blur kernel k. An illustrative example is shown in Figure 1.

In our experiments, we first confirm previous results on the importance of using the right blur kernel during inference [7] and show that even state-of-the-art methods break down if the wrong blur kernel is assumed. We also demonstrate that the proposed conditioned models can handle a large set of varying blur kernels, while other methods would have to train models for each single blur kernel. Moreover, the conditioned model almost achieves the same performance as models solely trained for one particular blur kernel and significantly outperforms other learning-based models trained for a single blur kernel.

2. Related Work

The problem of upscaling images has a long history in computer vision. This field can be roughly divided into two main branches. First, we have super-resolution methods that rely on multiple images of the same scene, which are either aggregated from a multi-camera setup or a video sequence [8, 25]. Most methods in this branch rely on the sub-pixel accurate alignment of the images in order to infer the values of missing pixels in the high-resolution domain. In this paper, we focus on the second branch of super-resolution algorithms, which try to find a high-resolution image from a single low-resolution image, *i.e.*, single image super-resolution (SISR) [10, 11].

A further distinction of super-resolution algorithms can be done by differentiating between the *blind* and the *non-blind* setup. As already mentioned in the introduction, blind super-resolution algorithms typically try to infer both, the unknown blur kernel k and the high-resolution image h. This task is often addressed in two-step scheme. First, the unknown blur kernel k is estimated from l and then, a non-blind super-resolution algorithm is employed for doing the actual image upscaling with the given blur kernel. Such a scheme is also advocated for image deconvolution [17].

Non-blind super-resolution is a highly active field and numerous algorithms appear every year. An influential work is that of Glasner *et al*. [11] who exploit the assumption that small patches often re-occur within the same image and across different scales. A very recent extension of this type of super-resolution methods is proposed in [13], which even uses affine transformations to find re-occurring patches. While these models can incorporate any given blur kernel on the fly during inference and perform well on repetitive structures, they are not state-of-the-art on natural images [13] and are typically slow.

The currently best performing approaches for SISR are based on different machine learning principles to directly learn a mapping between the low- and the high-resolution domain from a large set of training images. Dictionary learning approaches [33, 34] train coupled dictionaries in both domains to effectively represent the training data. During inference, low-resolution images are coded with the low-resolution dictionary (*e.g.*, sparse coding [21]). Due to the coupling of both domains, the same code is re-used to reconstruct the high-resolution data with the high-resolution dictionary. Neighborhood embedding approaches [2, 3] go without learned dictionaries and directly use all the training data for coding. Based on these two approaches, several models emerged that replace the slow coding step with a locally-linear regression. Locality is established either via a flat codebook [28, 29] or a hierarchical structure in a random forest [24]. Dong *et al*. [6] present a convolutional neural network for SISR which is very fast during inference and also achieves good results, similar to [24, 28].

However, all these learning-based approaches focus on a restrictive setup where a single blur kernel is fixed for both, training and testing phases. This is unrealistic in real-world scenarios [18] because the blur kernel is typically different for every test image. Assuming a wrong blur kernel during inference can lead to poor results as shown in Figure 1, *c.f.* [7]. In the field of image deconvolution, this is actually the standard setup and many different approaches exist to handle varying blur kernels during inference. The current state-of-the-art for this task also employs learning-based models, *e.g.*, shrinkage-fields [22], cascaded regression tree fields [14, 23], or fields of experts models [4]. However, these models are typically tailored towards image deconvolution and are often hard to adapt to SISR. In this work, we address the scenario of varying blur kernels for image upscaling and extend existing SISR models to be conditioned on and effectively handle a given blur kernel during inference.

3. Conditioned Regression Models

In this section we propose our extensions to recent state-of-the-art methods in order to handle low-resolution images underlying different blur kernels within a single model. Thus, the task is to recover a high-resolution image h given the low-resolution image l and the blur kernel k, where k is different for each image. A straight-forward approach is to simply increase the number of the training images by convolving each training image with several different blur kernels, train the individual methods without any further modification and ignore k as input during inference. This assumes that the models inherently learn to differentiate blur kernels only from the low-resolution images and, thus, have to be regarded as *blind* super-resolution methods. In the remainder of this work we will name such methods with the suffix *AB* (all blur kernels). In contrast to this, we show how to extend existing methods to be conditioned on a blur kernel as an additional input. Such models will subsequently have the suffix *CAB* (conditioned; all blur kernels).

All methods have in common that they are trained on patches, or sub-windows. We define the i-th low-resolution training patch as $\mathtt{x}_\mathtt{L}^{(i)}$ and the corresponding blur kernel and high-resolution patch as $k^{(i)}$ and $\mathtt{x}_\mathtt{H}^{(i)}$, respectively. The set of high-resolution patches is defined as $\mathtt{X}_\mathtt{H} = \{\mathtt{x}_\mathtt{H}^{(i)}\}_{i=1}^N$. Similarly, the set of low-resolution patches and corresponding blur kernels are denoted as $\mathtt{X}_\mathtt{L} = \{\mathtt{x}_\mathtt{L}^{(i)}\}_{i=1}^N$ and $\mathtt{X}_k = \{k^{(i)}\}_{i=1}^N$, respectively.

3.1. The Basic Conditioning Model

The current state-of-the-art in non-blind SISR are learning based methods. To make the output space tractable, methods such as global regression (GR) [29], anchored neighborhood regression (ANR) [29], A+ [28], and super-resolution forests (SRF) [24], perform learning and inference on small patches $\mathtt{x}_\mathtt{L}^{(i)}$, rather than on the complete image. These methods all predict an estimate $\hat{\mathtt{x}}_\mathtt{H}^{(i)}$ of the high-resolution patch $\mathtt{x}_\mathtt{H}^{(i)}$ given the corresponding low-resolution patch $\mathtt{x}_\mathtt{L}^{(i)}$ of an image.

Further, they utilize the patches in vectorized form, *i.e.*, the input is of the form $\mathbb{R}^{h \cdot w}$ rather than $\mathbb{R}^{h \times w}$, with h and w being the height and the width of the patch, respectively. This distinction seems to be obvious, but it enables us to stack a vectorized blur kernel $k^{(i)}$ to the individual low-resolution patches $\mathtt{x}_\mathtt{L}^{(i)}$. This already gives us the basic form of conditioning those methods, both during training and inference. Instead of only utilizing the low-resolution samples $\mathtt{X}_\mathtt{L}$ for training, we concatenate the input samples with the blur kernels $k^{(i)}$. Therefore, the new input samples are given by $\mathtt{X}_{\mathtt{L},k} = \{\mathtt{x}_{\mathtt{L},k}^{(i)}\}_{i=1}^N$, with $\mathtt{x}_{\mathtt{L},k}^{(i)} = [\mathtt{x}_\mathtt{L}^{(i)}, k^{(i)}]^\top$.

Because only the input changes in the basic conditioning model the above mentioned methods can be used without further adjustments. We refer the reader to the individual works for more details on the training and inference procedures. In the next two sections we show how to exploit the special structure of SRF [24] and SRCNN [6] to condition them on a blur kernel k in a more effective way.

3.2. Conditioned Super-Resolution Forest

Super-resolution forests [24] is a recently proposed model achieving state-of-the-art results for SISR. The basic idea of SRF is to model the super-resolution problem directly as a locally linear regression problem, which can be seamlessly addressed with random regression forests having multi-variate linear leaf node models. We refer to [24] for more details. While SRF can also be conditioned by the basic model explained above, the tree structure of the random forest enables us to incorporate a blur kernel k more tightly. We especially focus on the quality measure Q that is utilized to greedily optimize the split function σ of a split node. In the original work of [24] this measure is defined as

$$Q(\sigma, \theta, \mathtt{X}_\mathtt{H}, \mathtt{X}_\mathtt{L}) = \sum_{c \in \{\mathrm{Le}, \mathrm{Ri}\}} |\mathtt{X}_\mathtt{L}^c| \, E(\mathtt{X}_\mathtt{L}^c, \mathtt{X}_\mathtt{H}^c) \,, \tag{2}$$

where $|\cdot|$ defines the cardinality of a set, θ are the parameters of the split function that divides the samples $\mathtt{X}_\mathtt{L}$ and targets $\mathtt{X}_\mathtt{H}$ into a left and right branch, with corresponding data $\mathtt{X}_{\{\mathtt{L},\mathtt{H}\}}^{\mathrm{Le}}$ and $\mathtt{X}_{\{\mathtt{L},\mathtt{H}\}}^{\mathrm{Ri}}$, respectively. Further, the purity measure E is defined as

$$E(\mathtt{X}_\mathtt{L}, \mathtt{X}_\mathtt{H}) = |\mathtt{X}_\mathtt{L}|^{-1} \sum_{i=1}^{|\mathtt{X}_\mathtt{L}|} ||\mathtt{x}_\mathtt{H}^{(i)} - \bar{\mathtt{x}}_\mathtt{H}||_2^2 + \kappa_1 ||\mathtt{x}_\mathtt{L}^{(i)} - \bar{\mathtt{x}}_\mathtt{L}||_2^2 \,, \tag{3}$$

where $\bar{\mathtt{x}}_\mathtt{H}$ and $\bar{\mathtt{x}}_\mathtt{L}$ are the means of the target (high-resolution) and input (low-resolution) samples, respec-

tively. As suggested in [24], the purity measure E thus corresponds to a reduction in variance on both domains (high- and low-resolution) and κ_1 is a trade-off parameter.

In this work, we extend the above described quality measure for conditioned regression by introducing an additional regularization on the blur kernel:

$$E_c(\mathtt{X}_\mathrm{L}, \mathtt{X}_k, \mathtt{X}_\mathrm{H}) = E(\mathtt{X}_\mathrm{L}, \mathtt{X}_\mathrm{H}) + \kappa_2 |\mathtt{X}_k|^{-1} \sum_{i=1}^{|\mathtt{X}_k|} ||k^{(i)} - \bar{k}||_2^2 \,. \tag{4}$$

Using this term we can enforce a reduction in variance in terms of the blur kernels. This follows the intuition that low-resolution patches $\mathtt{x}_\mathrm{L}^{(i)}$ generated by the same blur kernel k should reach similar leaf nodes. As a consequence, the leafs become purer, which typically results in a better reconstruction of the high-resolution patches.

Additionally, we extend the regression model in the leaf nodes to a conditioned regression

$$\hat{x}(\mathtt{x}_\mathrm{L}, k) = W_p \cdot \mathtt{x}_\mathrm{L} + W_k \cdot k = W_{p,k} \cdot \mathtt{x}_{\mathrm{L},k} \,, \tag{5}$$

where $\hat{x}$ is the estimated high-resolution patch. W_p and W_k are the weight matrices of the regression and can be combined to a single regression matrix $W_{p,k}$.

The training and inference of the conditioned SRF remain unchanged and we refer to [24] for more details.

3.3. Conditioned Super-Resolution CNN

Another method for SISR is the super-resolution CNN (SRCNN) [6], which is motivated by the recent success of deep learning [16]. The network consists of three convolutional layers, whereas the first two are followed by a rectifier linear unit ($\mathrm{ReLU}(x) = \max(0, x)$) [20]. Although the network is also trained on (larger) patches, the inference is conducted on the whole image. This avoids the averaging step of individually estimated high resolution patches.

The special structure of a CNN architecture is not suited to directly concatenate low-resolution patches $\mathtt{x}_\mathrm{L}^{(i)}$ with a blur kernel $k^{(i)}$. One possibility would be to rewrite the CNN in terms of a fully-connected network by replacing the convolutions with inner products. With this approach we would again have to infer the high-resolution image $\hat{h}$ on densely extracted patches and average them. Therefore, we propose another way to condition the SRCNN on an additional blur kernel input.

The SRCNN [6] computes an estimate of a high-resolution image $\hat{h}$ given a low-resolution image l as

$$F_1(l) = \mathrm{ReLU}(W_1 * \uparrow_b l + B_1) \,, \tag{6}$$

$$F_2(l) = \mathrm{ReLU}(W_2 * F_1(l) + B_2) \,, \tag{7}$$

$$\hat{h} = W_3 * F_2(l) + B_3 \,, \tag{8}$$

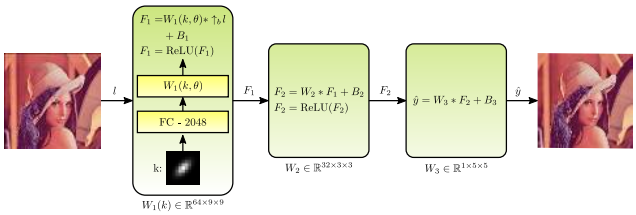


Figure 2: For our conditioned SRCNN we replace the first convolutional layer with a parameterized convolution. The constant convolution weights W_1 in the first layer are replaced with a non-linear function $W_1(k, \theta)$. This makes the filter weights dependent on an additional blur kernel input k. The function itself is realized via an extra, fully-connected neural network that is trained jointly with the SRCNN.

where F_i are multi-dimensional feature maps dependent on the low-resolution input and $\uparrow_b$ denotes a bicubic upsampling operation. W_i and B_i are convolutional and bias weights, respectively.

In a vanilla CNN, the weights of the convolutional filters are learned, but fixed at inference, independent of the input. We extend this formulation by replacing the constant weights with a parameterized weight function $W(k, \theta)$ that depends on learnable parameters θ and on some input k. Therefore, we change the definition of the first layer to

$$F_1(l) = \mathrm{ReLU}(W_1(k, \theta) * \uparrow_b l + B_1) \,. \tag{9}$$

The only difference to Equation (6) is the parameterization of the weights $W_1(k, \theta)$, which can now be an arbitrary function. A natural choice is to again utilize a neural network. This allows us to train the complete network end-to-end and let the network itself decide which filters are suitable for what kind of blur kernel. We employ a simple feed-forward network with a single hidden layer consisting of 2048 neurons. An overview of our complete network architecture is depicted in Figure 2.

For training the CNN we follow the procedure outlined in [6]. The network is implemented by extending the Caffe framework [15] for the parameterized convolution.

4. Experiments

In our experiments we analyze the effectiveness and performance of the proposed conditioned regression models for SISR on standard benchmarks. We first describe the experimental setup and the generation process of blur kernels before we present our main results on non-blind SISR.

4.1. Experimental Setup

We analyze and compare our proposed conditioned regression models for recent works in SISR, including global regression (GR) [29], anchored neighborhood regression

(ANR) [29], SRCNN [6], A+ [28], and super-resolution forests (SRF) [24]. For GR, ANR, and A+, we reuse the same settings as described in the papers [28, 29]. The SRF is trained with 10 trees to a maximal depth of 12, $\kappa_1 = 0.01$ and $\kappa_2 = 10$. The SRCNN is also parameterized as described in [6], except for our conditioned SRCNN (SRCNN CAB). In this case we utilize the proposed parameterized convolution in the first layer, see Section 3. If not stated otherwise, we employ the same set of parameters for all methods throughout the experiments. For training all our models, we always use a set of 91 images that were also used in [28, 29, 34]. For testing, we use the following standard benchmarks: `Set5` (5 images) and `Set14` (14 images) from [29, 34], and `BSDS` (200 images) [1].

All methods build upon the same framework for image upscaling: A given low-resolution image I_{low} is first converted into the YCbCr color space and is upscaled to the size of the desired high-resolution image via standard bicubic interpolation. While the resulting image I_{mid} already has the desired size, high-frequency components are missing and is thus referred to as mid-resolution. As the human visual perception is more sensitive to changes in intensity than in color [29], the high-frequency components I_{hf} are only added on the Y channel of the mid-resolution image. This yields the final estimate $I_{high} = I_{mid} + I_{hf}$. One exception is SRCNN [6] that directly predicts I_{high} from I_{mid}. The high-frequency estimate I_{hf} (or I_{high} for SRCNN [6]) is computed by the actual restoration model based on features (or raw data for SRCNN) extracted from I_{mid}.

For comparing the final output of different methods, we use the peak signal to noise ratio (PSNR), which is the most often used metric for such applications. We also include the IFC score [26], which has a much higher correlation with the human perception than PSNR [32].

4.2. Generating Blur Kernels for Super-Resolution

In order to evaluate the proposed non-blind super-resolution models, we require a set of realistic blur kernels for the image formation process. Existing methods to generate the ground truth blur kernels (*e.g.*, [9, 18, 19]) are typically slow and take several minutes per image, making them impractical to build a large corpus of training data. Here, we choose a Gaussian probability density function $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ with zero mean and varying covariance matrix $\mathbf{\Sigma}$ to represent the space of kernels. More precisely, we fix the size of the kernels to 11×11, allow rotation of the eigenvectors of $\mathbf{\Sigma}$ between 0 and π and scaling of the corresponding eigenvalues between 0.75 and 3. The rotation angle and the scaling fully define our space of blur kernels. Figure 3 shows 58 representative kernels uniformly sampled from this space.

While a single 2D anisotropic Gaussian may seem too simplistic, it is a reasonable assumption for the super-resolution problem (in contrast to image deblurring or deconvolution). When looking at [19], we see that most estimated kernels are actually unimodal and can typically be modeled with a Gaussian. We also note a recent SISR benchmark addressing the issue of different blur kernels [32]. While this benchmarks still assumes a fixed-blur-kernel during testing, it models the kernels only as a 2D isotropic Gaussian, which is already considered reasonable for this task. Finally, we should also note that all models presented in Section 3 are learning-based and not restricted to or modeled towards these kind of kernels. Thus, if other distributions of blur kernels prove to be more reasonable, it is easy to re-train our models for the new distribution.

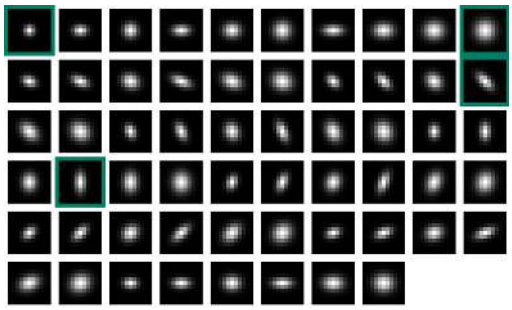


Figure 3: Visualization of the 58 blur kernels used in our experiments (normalized for better presentation). The kernels used in Section 4.3 are highlighted with a green border.

(a) Original (b) SRF, k_1: 34.52, 7.79

(c) SRF, k_2: 25.99, 4.25 (d) SRF-CAB: 34.13, 7.03

Figure 4: (a) depicts the high-resolution image h. The corresponding low-resolution image l was created with k_1 (top left corner). (b-c) show the up-sampling result obtained by the SRF, which was trained on k_1 and k_2, respectively. (d) presents the result of our proposed conditioned SRF. The image is taken from [24] and the upscaling factor is 2.

4.3. Learning for the Right Blur Kernel is Essential

In this section we highlight the importance of learning for the right blur kernel, thus extending previous results presented by Efrat *et al.* [7] with recent state-of-the-art SISR

Method		k_1	k_2	k_3	k_4
ANR [29]	k_1	**35.45**	29.74	31.95	32.07
	k_2	26.38	**32.78**	29.93	29.35
	k_3	30.35	31.20	**34.10**	31.61
	k_4	30.55	31.03	31.84	**33.95**
A+ [28]	k_1	**36.22**	29.66	32.03	32.15
	k_2	25.89	**33.46**	29.36	29.19
	k_3	29.56	31.29	**34.96**	31.41
	k_4	29.70	31.09	31.31	**34.82**
SRCNN [6]	k_1	**36.29**	29.72	32.17	32.26
	k_2	15.51	**35.47**	18.35	22.50
	k_3	26.30	31.64	**35.68**	29.73
	k_4	23.69	31.60	27.80	**35.93**
SRF [24]	k_1	**36.24**	29.75	32.13	32.24
	k_2	25.52	**33.66**	29.33	28.97
	k_3	29.72	31.41	**34.97**	31.51
	k_4	29.75	31.27	31.60	**34.88**
SRCNN-CAB	-	32.38	32.56	33.20	33.28
SRF-CAB	-	*35.40*	*32.89*	*34.10*	*34.09*

Table 1: Results for `Set5` and upscaling factor 2 when using different blur kernels during training and evaluation.

methods. We select four different kernels, highlighted green in Figure 3, which are representative for our kernel space.

In Table 1 we compare four state-of-the-art methods (ANR [29], A+ [28], SRCNN [6], and SRF [24]) and train each method on the 4 selected blur kernels $k_{1,\dots,4}$. We thus get 16 models corresponding to the first 16 rows in the table. Then, we evaluate each model four times on `Set5`, each time blurred with one of the 4 kernels $k_{1,\dots,4}$ and downsampled. The results are stated as mean PSNR value for an upscaling factor of 2 in the columns of the table. We can observe the expected result: Models trained for the blur kernel also used during testing achieve the best results. However, we can also see that the results *significantly* drop if we evaluate a method trained on k_i and evaluate it on images convolved with k_j ($i \neq j$). An illustrative example is SRCNN, which achieves the best overall results but also shows the most pronounced score drops for different kernels.

For comparison, we include results of two of our extensions that exploit the full set of blur kernels in a single model (CAB). These models are described in Section 3 in more detail. As expected, non of these models outperform the specialized trained models, but they are significantly better than models trained with a different blur kernel. For instance, SRF [24] does not achieve a PSNR value above 30 dB when testing on k_1, except the exact same blur kernel was used for training. In contrast, SRF-CAB achieves 35.40 dB. We also illustrate this behavior in Figure 4.

4.4. Non-Blind Single Image Super-Resolution

Here, we present our main results on non-blind super-resolution. Again, the task is to compute a visually pleasing high-resolution image h given its low-resolution counterpart l and the corresponding blur kernel k used to create the low-resolution image. The blur kernel is thus *known but different* for each test image, unlike traditional work on SISR that uses a fixed kernel for training and inference.

As already stated in Section 2, related work for this specific task is actually rather limited because most of the state-of-the-art super-resolution approaches are trained and evaluated for a single blur kernel. However, we can still use recently proposed models ([6, 24, 28, 29]) in our setup by adapting the training procedure as described in Section 3. We used the 91 training images described above and randomly sampled 14 different blur kernels for each training image (out of the 58 from Figure 3) to create the corresponding low-resolution image. This results in a training set of 1274 low- and high-resolution image pairs.

Using this data, we can train our extensions of the state-of-the-art methods (AB and CAB) as described in Sections 3.1, 3.2, and 3.3. For each training patch, we collect the corresponding blur kernel in a vectorized form and, similar to the image features in the framework of [29], we first apply PCA on the blur kernels. This reduces the dimensionality from $11 \times 11 = 121$ to only 8 while still preserving 99% of the energy. This 8-dimensional vector representing the blur kernel is stacked with the low-resolution patches, allowing each method to access the current blur kernel of the image during both training and testing.

As baselines, we include models that are trained with the assumption of a bicubic blur kernel. We also note that we do not include non-blind super-resolution methods based on the patch re-occurrence assumption like [11, 13], which can incorporate any blur kernel during inference, as we do not have access to a proper implementation. A re-implementation is prone to errors and could lead to suboptimal results. However, we refer to the very recent results of [13], which demonstrate impressive results on images with highly repetitive structures, but are inferior on natural images (`BSDS`) compared to standard A+ [28]. We further do not include any adaption of recent related deconvolution methods ([4, 14, 23, 22]). Although some methods would, in therory, only require in addition to the blur kernel matrix a down-sampling matrix, methods like [22] crucial rely on the specific structure of the blur kernel matrix, which is not given if the down-sampling matrix is included. A simple adaptation is therefore often not feasible. Still, modifying recent condition regression models like [5] would be an interesting research direction.

Table 2 depicts our quantitative results for the 3 different test sets and different upscaling factors (2, 3, and 4). We report the mean PSNR and IFC scores over all test images, convolved with all 58 blur kernels from Figure 3, which results in 290, 812, and 11,600 test images for `Set5`, `Set14`, and `BSDS`, respectively. If the model was only trained for a single blur kernel (bicubic; Bic), we can generally observe inferior results compared to models

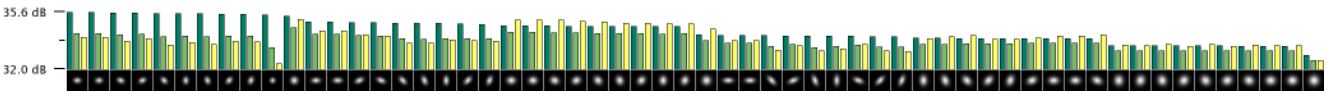


Figure 5: Complexity of individual blur kernels for SRF-CAB (dark-green), A+-CAB (light-green), and SRCNN-CAB (yellow). The bars show the mean PSNR value over the images of `Set5` (upscaling factor 2) for the individual 58 blur kernels.

Method		Set5			Set14			BSDS	
		x2	x3	x4	x2	x3	x4	x2	x3
GR	Bic	29.32/2.86	26.63/1.71	24.38/1.08	27.04/2.80	24.88/1.57	23.04/0.95	27.01/2.62	25.12/1.42
ANR	Bic	29.35/2.85	26.65/1.72	24.43/1.09	27.06/2.78	24.88/1.57	23.09/0.96	27.01/2.61	25.13/1.42
A+	Bic	29.34/2.85	26.60/1.73	24.27/1.09	27.05/2.78	24.82/1.57	22.92/0.94	27.01/2.61	25.07/1.42
SRCNN	Bic	29.38/2.86	26.60/1.71	24.29/1.06	27.09/2.79	24.82/1.56	22.95/0.93	27.04/2.62	25.09/1.41
SRF	Bic	29.36/2.85	26.65/1.74	24.35/1.09	27.07/2.78	24.87/1.58	23.00/0.95	27.03/2.60	25.12/1.43
GR	AB	32.80/5.14	30.72/3.47	28.85/2.32	29.63/5.11	27.78/3.26	26.26/2.08	29.11/4.86	27.36/3.01
ANR	AB	33.06/5.15	31.14/3.65	29.26/2.49	29.77/5.07	28.05/3.39	26.57/2.21	29.16/4.80	27.51/3.10
A+	AB	33.21/4.90	31.30/3.59	29.37/2.45	30.00/4.78	28.23/3.30	26.71/2.15	29.34/4.53	27.62/3.02
SRCNN	AB	33.58/5.31	31.43/3.66	29.50/2.51	30.27/5.25	28.33/3.41	26.76/2.20	29.62/5.01	27.70/3.13
SRF	AB	33.50/5.24	31.44/3.74	29.38/2.51	30.11/5.12	28.24/3.45	26.68/2.22	29.41/4.83	27.63/3.15
GR	CAB	32.78/5.11	30.72/3.47	28.85/2.33	29.61/5.08	27.78/3.27	26.26/2.09	29.09/4.84	27.36/3.01
ANR	CAB	32.86/4.87	31.09/3.57	29.22/2.47	29.60/4.77	28.01/3.32	26.54/2.20	28.98/4.47	27.48/3.02
A+	CAB	33.76/*5.50*	31.35/3.65	29.27/2.40	30.35/5.37	28.28/3.37	26.63/2.11	29.65/5.09	27.67/3.07
SRCNN	CAB	*33.92*/5.41	*31.70*/*3.78*	*29.54*/*2.55*	*30.50*/*5.42*	*28.49*/*3.51*	*26.81*/*2.23*	*29.82*/*5.17*	*27.81*/*3.21*
SRF	CAB	**34.43/6.06**	**31.91/4.12**	**29.64/2.63**	**30.73/5.87**	**28.59/3.77**	**26.86/2.31**	**29.90/5.53**	**27.89/3.44**

Table 2: Quantitative super-resolution results for three benchmarks. We present the mean PSNR and IFC scores for each test set, where all images of the test set are convolved with each of the 58 blur kernels. The best results are highlighted **bold-face**, and the second best *italic*. Results for `BSDS` x4 are included in the supplemental material.

trained with all 58 kernels, even the ones in the *blind* super-resolution setting, *i.e.*, AB. Whether or not the basic conditioning on the blur kernel improves strongly, depends on the method. For GR and ANR the results remain nearly unchanged, or get even slightly worse for the conditioned model. On the other hand, A+ already benefits from the simple conditioning model, and we can also observe that our proposed extensions, SRCNN-CAB and SRF-CAB, improve clearly over the AB methods. We also present some qualitative results in Figure 7.

In Figure 5, we demonstrate the influence of the individual blur kernels on the results for the three best-performing methods, evaluated on `Set5` and an upscaling factor of 2. The SRF-CAB performs clearly better if the blur kernel has a small spatial extent or if it is elongated. In contrast, the SRCNN-CAB peaks for medium-sized blur kernels and all models get worse if the kernel exceeds a certain size.

4.5. Generalizing to Unseen Blur Kernels

In the previous experiment, we evaluated several models for varying blur kernels. For a proper investigation of the complexity of the blur kernels, we used the same set of 58 kernels for both, training and testing phase. In practical applications, though, we are interested in upscaling images generated with previously unseen kernels, *i.e.*, generalization. In this section, we investigate this issue for the best performing model from the previous experiment, *i.e.*, SRF-CAB. We thus train 7 models with access to only a fraction of the 58 blur kernels from Figure 3 (58, 29, 15, 8, 4, 2, 1). During inference, all models still have to handle all 58 kernels. We show our results in Figure 6, where the 7 bars above each kernel depict the differently trained models. The most left one has access to all kernels, while the most right one uses only a single kernel during training. The red dots below the bars indicate whether or not this particular model had access to the kernel below. The results indicate that SRF-CAB can handle previously unseen blur kernels during testing. While using only a single kernel to train the model gives poor performance, only a small fraction of the kernels already suffices to achieve good overall performance.

4.6. Analyzing the Conditioned Regression Forest

As SRF-CAB turned out to give the best overall results in Section 4.4, we analyze this model in more detail.

In a first experiment, we investigate the influence of the blur kernel to different parts of the forest. We train the random forest in four different settings: (a) blur kernels are not used at all except for generating the training data, *i.e.*, the AB model; (b) blur kernels are only used on the split nodes (B split); (c) blur kernels are only used in the leaf nodes (B leaf); (d) blur kernels are always available as described in Section 3, *i.e.*, the CAB model. Figure 8a depicts our results as PSNR on `Set5` for an upscaling factor of 2. We can observe that conditioning on the blur kernel always improves

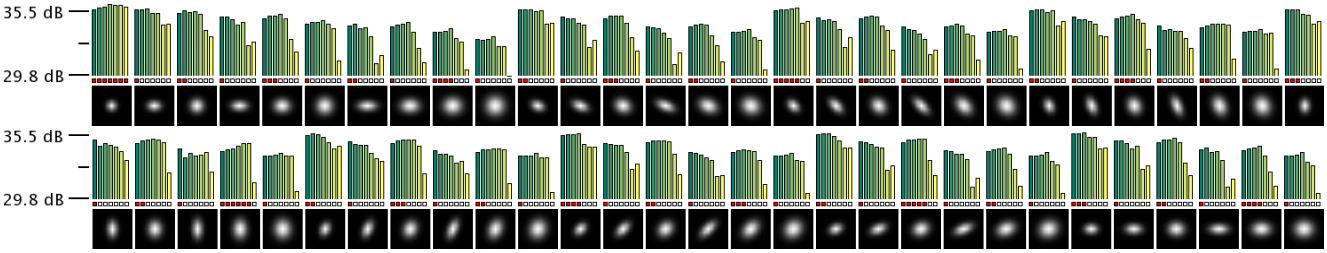
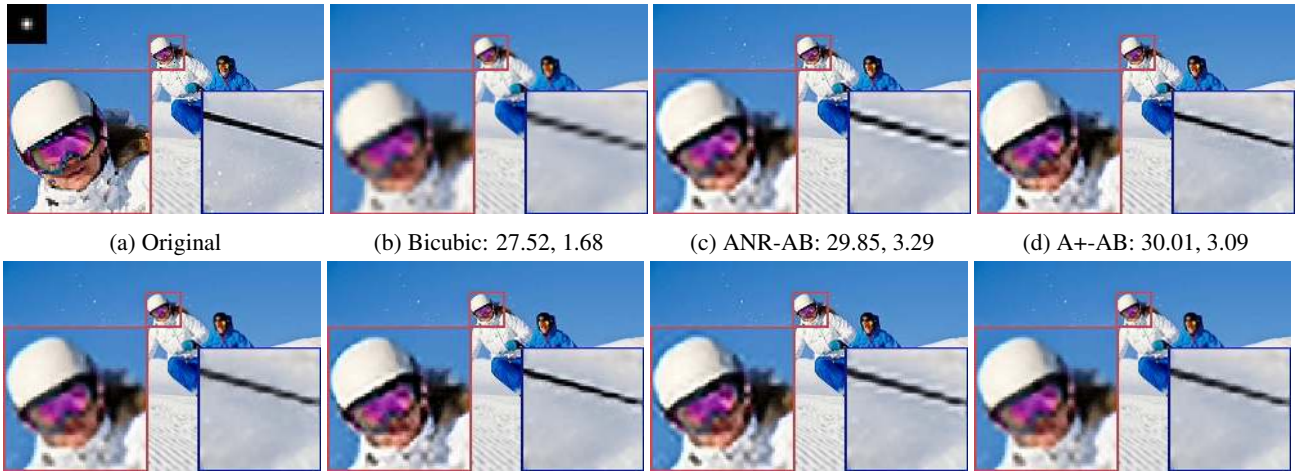


Figure 6: Generalization power of 7 SRF-CAB model, each trained with access to a different amount of blur kernels. The figure shows the results (PSNR, `Set5`) of the models on all 58 blur kernels (separated into two rows). See text for details.

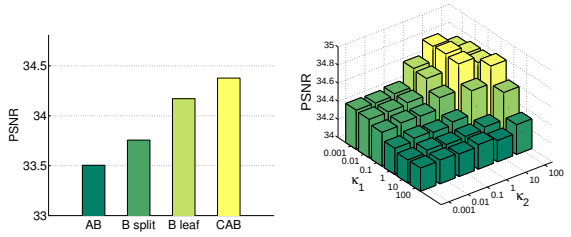


(a) Original (b) Bicubic: 27.52, 1.68 (c) ANR-AB: 29.85, 3.29 (d) A+-AB: 30.01, 3.09

(e) SRF-AB: 30.20, 3.22 (f) A+-CAB: 30.08, 2.96 (g) SRCNN-CAB: 30.03, 3.22 (h) SRF-CAB: 30.48, 3.33

Figure 7: Qualitative results of state-of-the-art methods for upscaling factor x3 on image `skiing` (image taken from [24]). The numbers in the sub-captions refer to PSNR and IFC scores, respectively. Best viewed in color and digital zoom.

the results, regardless of what part in the forest has access to it. The biggest gain is obtained by utilizing the blur kernel in the leaf nodes, but fully conditioning on the blur (CAB) yields the best results. In a second experiment, we evaluate the influence of κ_1 and κ_2 on the PSNR scores on `Set5` for an upscaling factor of 2. Figure 8b depicts our results. We can observe a good performance with a small regularization of the low resolution patches (κ_1) and a high regularization of the blur kernels (κ_2). More details can be found in the supplemental material.



(a) (b)

Figure 8: (a) Influence of the blur kernel for different parts of the forest. (b) Evaluation of the parameters κ_1 and κ_2.

5. Conclusion

Previous state-of-the-art methods in SISR typically rely on machine-learning techniques that learn a mapping from low- to high-resolution images. Although achieving good results, these methods are trained and evaluated for just a single blur kernel. For practical applications, however, the blur kernel is typically different for each test image. In this work, we tackle this problem by proposing several conditioned regression models that can incorporate different blur kernels during training and inference.Beside demonstrating the importance of using the right blur kernel for each test image, our experiments also show the effectiveness of the proposed conditioned models on an extensive evaluation. When evaluated for a single blur kernel, the conditioned models almost achieve the same performance as models specifically trained for that particular kernel. In contrast, when evaluating for different blur kernels, the conditioned models show their effectiveness and outperform other methods that are only trained for a single fixed blur kernel.

We hope that future research in non-blind SISR favors the proposed evaluation setup with varying blur kernels over the prevailing assumption of a single fixed blur kernel.

Acknowledgment: This work was supported by the Austrian Research Promotion Agency (FFG) projects TOFUSION (FIT-IT Bridge program, #838513) and AIRPLAN (Kiras program, #840858).

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour Detection and Hierarchical Image Segmentation. *PAMI*, 33(5):898–916, 2011. 5

[2] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. Alberi Morel. Low-Complexity Single-Image Super-Resolution based on Nonnegative Neighbor Embedding. In *BMVC*, 2012. 2

[3] H. Chang, D.-Y. Yeung, and Y. Xiong. Super-Resolution Through Neighbor Embedding. In *CVPR*, 2004. 2

[4] Y. Chen, T. Pock, R. Ranftl, and H. Bischof. Revisiting loss-specific training of filter-based MRFs for image restoration. In *GCPR*, 2013. 3, 6

[5] Y. Chen, W. Yu, and T. Pock. On Learning Optimized Reaction Diffusion Processes for Effective Image Restoration. In *CVPR*, 2015. 6

[6] C. Dong, C. Change Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *ECCV*, 2014. 1, 2, 3, 4, 5, 6

[7] N. Efrat, D. Glasner, A. Apartsin, B. Nadler, and A. Levin. Accurate Blur Models vs. Image Priors in Single Image Super-Resolution. In *ICCV*, 2013. 2, 3, 5

[8] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar. Fast and Robust Multiframe Super Resolution. *TIP*, 13(10):1327–1344, 2004. 2

[9] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing Camera Shake from a Single Photograph. In *SIGGRAPH*, 2006. 5

[10] W. T. Freeman, T. R. Jones, and E. C. Pasztor. Example-Based Super-Resolution. *CGA*, 22(2):56–65, 2002. 1, 2

[11] Glasner, Daniel, Bagon, Shai and Irani, Michal. Super-Resolution From a Single Image. In *ICCV*, 2009. 2, 6

[12] S. Harmeling, S. Sra, M. Hirsch, and B. Schölkopf. Multiframe Blind Deconvolution, Super-Resolution, and Saturation Correction via Incremental EM. In *ICIP*, 2010. 2

[13] J.-B. Huang, A. Singh, and N. Ahuja. Single Image Super-Resolution using Transformed Self-Exemplars. In *CVPR*, 2015. 2, 6

[14] J. Jancsary, S. Nowozin, T. Sharp, and C. Rother. Regression Tree Fields. In *CVPR*, 2012. 3, 6

[15] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional Architecture for Fast Feature Embedding. *arXiv preprint arXiv:1408.5093*, 2014. 4

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*, 2012. 4

[17] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *CVPR*, 2009. 2

[18] T. Michaeli and M. Irani. Nonparametric Blind Super-Resolution. In *ICCV*, 2013. 1, 2, 3, 5

[19] T. Michaeli and M. Irani. Blind Deblurring Using Internal Patch Recurrence. In *ECCV*, 2014. 5

[20] V. Nair and G. E. Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. In *ICML*, pages 807–814, 2010. 4

[21] B. A. Olshausen and D. J. Field. Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by V1? *VR*, 37(23):3311–3325, 1997. 2

[22] U. Schmidt and S. Roth. Shrinkage Fields for Effective Image Restoration. In *CVPR*, 2014. 3, 6

[23] U. Schmidt, C. Rother, S. Nowozin, J. Jancsary, and S. Roth. Discriminative Non-blind Deblurring. In *CVPR*, 2013. 3, 6

[24] S. Schulter, C. Leistner, and H. Bischof. Fast and Accurate Image Upscaling with Super-Resolution Forests. In *CVPR*, 2015. 1, 2, 3, 4, 5, 6, 8

[25] Shechtman, Eli and Caspi, Yaron and Irani, Michal. Space-Time Super-Resolution. *PAMI*, 27(4):531–545, 2005. 2

[26] H. R. Sheikh, A. C. Bovik, and G. De Veciana. An information fidelity criterion for image quality assessment using natural scene statistics. *TIP*, 14(12):2117–2128, 2005. 1, 5

[27] F. Sroubek, G. Cristobal, and J. Flusser. Simultaneous Super-Resolution and Blind Deconvolution. In *Journal of Physics: Conference Series*, 2008. 2

[28] R. Timofte, V. De Smet, , and L. Van Gool. A+: Adjusted Anchored Neighborhood Regression for Fast Super-Resolution. In *ACCV*, 2014. 1, 2, 3, 5, 6

[29] R. Timofte, V. De Smet, and L. Van Gool. Anchored Neighborhood Regression for Fast Example-Based Super-Resolution. In *ICCV*, 2013. 2, 3, 4, 5, 6

[30] M. Unger, T. Pock, M. Werlberger, and H. Bischof. A Convex Approach for Variational Super-Resolution. In *DAGM*, 2010. 1

[31] Q. Wang, X. Tang, and H. Shum. Patch Based Blind Image Super Resolution. In *ICCV*, 2005. 2

[32] C.-Y. Yang, C. Ma, and M.-H. Yang. Single-Image Super-Resolution: A Benchmark. In *ECCV*, 2014. 5

[33] J. Yang, J. Wright, T. Huang, and Y. Ma. Image Super-Resolution Via Sparse Representation. *TIP*, 19(11):2861–2873, 2010. 2

[34] R. Zeyde, M. Elad, and M. Protter. On Single Image Scale-Up using Sparse-Representations. In *Curves and Surfaces*, 2010. 2, 5