

Supervised Deep Kriging for Single-Image Super-Resolution

Gianni FRANCHI¹, Angela YAO², and Andreas KOLB¹

¹ Institute for Vision and Graphics, University of Siegen, Germany
{gianni.franchi, andreas.kolb}@uni-siegen.de

² University of Bonn, Germany
yao@cs.uni-bonn.de

Abstract. We propose a novel single-image super-resolution approach based on the geostatistical method of kriging. Kriging is a zero-bias minimum-variance estimator that performs spatial interpolation based on a weighted average of known observations. Rather than solving for the kriging weights via the traditional method of inverting covariance matrices, we propose a supervised form in which we learn a deep network to generate said weights. We combine the kriging weight generation and kriging process into a joint network that can be learned end-to-end. Our network achieves competitive super-resolution results as other state-of-the-art methods. In addition, since the super-resolution process follows a known statistical framework, we are able to estimate bias and variance, something which is rarely possible for other deep networks.

1 Introduction

Super-resolution aims to transform low resolution (LR) images into images with high resolution (HR). In computer vision, super-resolution is relevant for applications where high-frequency information and detailing is desirable yet not always captured, *e.g.* medical imaging, satellite imaging, surveillance, etc. Our interest is in single image super-resolution (SISR), a special case where only one LR image is available. SISR is an ill-posed inverse problem with multiple solutions, *i.e.* multiple HR images could lead to the same LR image after applying a low-resolution filter.

SISR been solved with many different approaches. Early approaches relied on spatial relationships present in the image and include nearest neighbours [4], bi-cubic interpolation [13]. Other techniques build upon the assumption that the underlying signal is sparse, *e.g.* compressive sensing [8] and dictionary learning [30]. These early approaches tend to yield HR images which are blurry and or noisy, due mainly to the fact that these methods are crafted heuristically. As a result, they cannot sufficiently recover enough information to yield sharp photo-realistic images.

Deep learning, and more specifically, convolutional neural networks (CNNs) [16], has afforded significant performance gains in SISR. The newest frameworks [14,15,17,25,24] learn feature representations for super-resolution in a supervised

end-to-end manner and are achieving ever-improving performance both in terms of speed and quantitative evaluation. However, despite their impressive results, most deep networks are being used as black boxes to make point estimates. The regressed outputs represent only a mean value and give no indication of model uncertainty. In the case of SISR, this means that we make (possibly very accurate) guesses for the unknown pixels, but we have no idea how good these guesses are. This can be highly problematic for applications such as microscopy or medical imaging, where model confidence is just as important as the actual output.

We would like to leverage the strong learning capabilities of deep networks and embed them within a known statistical framework for which we can derive model uncertainty. As such, we propose *deep kriging* to solve SISR. Kriging [20,5] is a geostatistics method used for spatial interpolation of attributes such as topography and natural resources. It has close relations to Gaussian process (GP) regression, although some sources oversimplify and describe the two as being equivalent. The main difference is that kriging makes stationary and ergodic assumptions on the random field representing the data, while GP regression considers it as a Gaussian process.³

Kriging interpolates unknown values by taking a weighted average of known values and results in an unbiased estimator with minimum variance. One can solve for the weights by inverting a covariance matrix computed on the known instances. Kriging makes for a natural extension to the application of SISR, where HR images are interpolated from pixels observed in the LR version of the image. The computational cost of the kriging is cubic with respect to the number of known instances; hence it is preferable to work locally. However, interpolated results are often overly smooth and depend on the choice of the covariance function.

Our contributions can be summarized as follows:

- We propose a novel deep learning-based statistical estimator with SISR results competitive with state-of-the-art.
- We propose a deep kriging approach that solves for kriging weights and performs the kriging in a single network; this allows us to perform a supervised form of spatial interpolation which can be applied not only to SISR, but other geostatistical computations such as field reconstructions.
- Our proposed deep kriging is a hybrid deep learning and statistical framework for which we can derive statistical bias and variance; such measures of model performance and uncertainty are not commonly available for other deep networks, making us the first to model and compute pixel uncertainty in deep SISR approaches.

2 Related work

Prior to the use of deep learning, SISR approaches applied variants of dictionary learning [9,8,30,22,7]. Patches were extracted from the low-resolution images and

³ We refer the reader to our supplementary materials for a more detailed comparison of the two.

mapped to their corresponding high-resolution version which are then stitched together to increase the image resolution. Other learning-based approaches to increase image resolution include [33,32,1].

State-of-the-art SISR methods are deep-learning-based [17,14,15,24,25]. The VDSR [14] and DRCN [15] approaches showed the benefits of working with image residuals for super-resolution. The DRRN approach [25] generalizes VDSR and concludes that the deeper the network, the better the super-resolved image. We also use a residual network in our approach, but unlike all the other deep SISR methods, we are solving for a set of filter weights to perform the super-resolution with our network rather than directly estimating the HR image itself.

Several unsupervised techniques have also been developed for SISR [29,18], though their performance is usually poorer than supervised approaches. One work of note [10], uses GP regression to perform super-resolution and resembles ours in spirit in that we both model pixel intensities as a random process regressed from neighbouring pixels. However, [10] is unsupervised while our approach learn the weights from supervise examples.

Our proposed approach can be thought of as a form of local filtering [3], and is most similar in spirit to works which combine deep learning and GP regression [28,6]. However, we differ from these techniques since we do not apply GP regression but a modified version of local kriging. These techniques are similar to us in that they also consider the data to follow a random process. However, we do not explicitly learn the covariance matrix, which offers us more flexibility in the relationships we wish to express. Furthermore, we treat each image as following a random process; the set of training images is then a set of random processes, while the aforementioned works consider all the training data to follow the same process.

3 Deep Kriging

We begin with a short overview on classical kriging in Section 3.1 before we introduce our proposed method of supervised kriging in Section 3.2 and elaborate on its statistical properties in Section 3.3. Finally, we show how the proposed form of deep kriging can be implemented in a deep network in Section 3.4.

3.1 Classical kriging

Consider an image f as a realization of a random field F . This random field is a collection of random variables, *i.e.* $F = \{F_i\}$ where each F_i is a random variable and i is a position index. Unknown values on f at position x^* , \hat{f}_* ⁴, can be interpolated linearly from n known realizations f_i , with normalized weights w_i , *i.e.*

$$\hat{f}_* = \sum_{i=1}^n \omega_i(x^*) f_i, \quad \text{with} \quad \sum_{i=1}^n \omega_i(x^*) = 1. \quad (1)$$

⁴ For concise notation, we use \hat{f}_* to denote $\hat{f}(x^*)$, f_i to denote $f(x_i)$ and $w_i^* = w_i(x^*)$.

In classical kriging, all known realizations are used in the interpolation and n is the number of pixels in the image, though local variants have been proposed and studied in [23,21].

The weights $\{\omega_i\}$ are found by minimizing the true risk:

$$R(\omega) = \mathbb{E} \left[(F_* - \hat{F}_*)^2 \right] = \text{var} \left(F_* - \hat{F}_* \right). \quad (2)$$

We can equate the $\mathbb{E}[\cdot]$ with the $\text{var}(\cdot)$ term in Eq. 2 because the constraint that the weights must sum up to 1 implies that $\mathbb{E}[f_* - \hat{f}_*] = 0$. However, since we do not have access to different realizations of the random field, this variance cannot be solved directly.

To infer the variance from a single event, the theory of geostatistics replaces the classical statistics assumption of having independent and identically distributed (iid) random variables with assumptions of stationarity and ergodicity on the random field. The random field is assumed to be first and second order stationary. This means that $\mathbb{E}_{F^i}(f^i(x))$ does not depend on x and that $\mathbb{E}_{F^i}(f^i(x) \times f^i(x+\tau))$ depends only on τ . In addition, the field is assumed to be second order ergodic, so a covariance or mean estimate in the probability domain is equivalent in the spatial domain. This second assumption implies that an empirical covariance can be estimated and depends only on a distance τ between two realizations, *i.e.* $\tau = \|x_i - x_j\|$. However, it is difficult to work directly with the empirical covariance, since it can be noisy and also is not guaranteed to form a positive semi-definite matrix. As such, in kriging, we use the true covariance by fitting a parametric model. While different models can be used, the most common one is Gaussian, with covariance C_{ij} between points x_i and x_j defined as $C_{ij} = C_0 \exp(-\frac{1}{\sigma^2} \|x_i - x_j\|^2)$, where C_0 and σ are parameters of the model.

A Lagrange multiplier (with constant λ) can be used to minimize the risk function with the constraints on the weights w_i , leading to the following cost function:

$$\mathcal{L}(w) = \text{var} \left(F_* - \hat{F}_* \right) + 2\lambda \left(\sum_{i=1}^n w_i(x^*) - 1 \right), \quad (3)$$

and the associated solution expressed in matrix form as:

$$\begin{pmatrix} C_{11} & \dots & C_{1n} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ C_{1n} & \dots & C_{nn} & 1 \\ 1 & \dots & 1 & 0 \end{pmatrix} \begin{pmatrix} w_1(x^*) \\ \vdots \\ w_n(x^*) \\ -\lambda \end{pmatrix} = \begin{pmatrix} C_{1*} \\ \vdots \\ C_{n*} \\ 1 \end{pmatrix}. \quad (4)$$

The process of kriging then is reduced to solving for the weights w_i . This involves inverting the covariance matrix on the left. Since the matrix is of dimension $(n+1)$, where n is the number of observed pixels in the image (patch), it can be very computationally expensive. Even though one can limit n with local kriging, n needs to be sufficiently large in order to accurately capture and represent the spatial relationships. We aim to bypass the matrix inversion and still maintain sufficient generalization power by using a deep network to directly learn the weights instead.

3.2 Supervised kriging

Consider E , a subset of the discrete space \mathbb{Z}^2 , as a representation of the support space of a 2D image. We denote an image as $f \in \mathbb{R}$ and f_i as its value at pixel $x_i \in E$. This assumes we work with greyscale images, which is a standard assumption made in most SISR algorithms [15, 14, 25].

We assume that we are given a set of n_1 training image pairs $\{(\tilde{f}^i, f^i)\}$, $i \in [1, n_1]$, where \tilde{f}^i is an up-sampled low resolution version of f^i with the same size. Our objective is to super-resolve the low-resolution images of the test set $\mathcal{N}_{\text{test}} = \{\tilde{f}^i\}_{i=1}^{n_2}$. We further assume that \tilde{f}^i is a realization of a first and second moment stationary random process \tilde{F}^i . For convenience we denote \tilde{F}^i the distribution of the random process. In addition f is a realization of a random field F . We denote the training set as $\{\tilde{f}^i, f^i\}_{i=1}^{n_1} \sim \mathcal{M}$, where \mathcal{M} is a joint meta-distribution where each training pair $(\tilde{f}^i, f^i) \sim \tilde{F}^i \times F^i$ follows its own random process.

In classical kriging, the estimated $\hat{f}(x)$ is expressed as a linear combination of all the observed samples *i.e.* pixels of f . In the case of super-resolution, the observations come from the low-resolution image $\tilde{f}(x)$. Furthermore, we estimate $\hat{f}(x)$ as a linear combination of only local observations of some window radius \mathcal{K} , leading to the estimator:

$$\hat{f}_*^i = \sum_{k \in \{\|x_k - x^*\|_1 \leq \mathcal{K}\}} \omega_k(x^*) \tilde{f}_k^i \quad \text{with} \quad \sum_{k \in \{\|x_k - x^*\|_1 \leq \mathcal{K}\}} \omega_k(x^*) = 1. \quad (5)$$

When no confusion is possible we denote $\hat{F}_*^i = \sum_{k=1}^n \omega_k(x^*) \tilde{F}_k^i$, with $n = (2\mathcal{K} + 1)^2$. We found that a window of 7×7 provided the best results. To find ω , we want to minimize the true risk as given in Eq. 2, leading to:

$$R(\omega) = \sum_{i=1}^{n_1} \mathbb{E}_{F^i \times \tilde{F}^i} \left[\left(F_*^i - \hat{F}_*^i \right)^2 \right]. \quad (6)$$

This risk is a compromise between the geostatistical model and the deep learning model, where the covariance of each field is learned through supervised learning. Furthermore, if we replace the true expectation with the empirical one and the random field with its realization, we arrive at

$$R(\omega) = \sum_{i=1}^{n_1} \sum_{j=1}^{N_i} \left[\left(f_*^i - \sum_{k=1}^n \omega_k(x^*) \tilde{f}_k^i \right)^2 \right], \quad (7)$$

where N_i is the number of pixels in image f^i .

Now, rather than solving for the weights ω by inverting the covariance matrix in Eq. 4, we propose learning this set of weights directly with a CNN. This allow us to capture and model more complex spatial relationships between the data than the classical model with covariances. As such, the estimator becomes $\hat{f}(x, \tilde{f}, g, \theta)$, where $w = g(\tilde{f}, \theta)$, with g and θ representing the CNN network function and its parameters respectively.

3.3 Statistical properties

We can derive statistical properties for our proposed estimator \hat{F} . Given $\{\omega_1, \dots, \omega_n\}$, the bias can be expressed as

$$\begin{aligned} \text{bias}(x) &= \mathbb{E}_{\tilde{F}|\omega_1, \dots, \omega_n} [\tilde{F}(x) - \hat{F}(x)] \\ &= \mathbb{E}_{\tilde{F}|\omega_1, \dots, \omega_n} [\tilde{F}(x)] - \mathbb{E}_{\tilde{F}|\omega_1, \dots, \omega_n} [\hat{F}(x)] \\ &= \mathbb{E}_{\tilde{F}|\omega_1, \dots, \omega_n} [\tilde{F}(x)] - \sum_{k_1=1}^n \omega_{k_1}(x_j) \mathbb{E}_{\tilde{F}|\omega_1, \dots, \omega_n} [\tilde{F}(x)]. \end{aligned}$$

Since $\sum_{k_1=1}^n \omega_{k_1}(x_j) = 1$ and the random field is first-order stationary, we are left with zero bias according to the weights of the neural network, *i.e.* $\text{bias}(x) = 0$. In other words, our network does not add bias to the field \tilde{F} . As such, from F to \tilde{F} , if the two fields have the same first moment, then no bias added to F ⁵. This point is critical since in classical statistics a good estimator should have zero bias and minimal variance.

By definition, the variance of our estimator \hat{F} is

$$V_{\hat{F}}(x) = \mathbb{E}_{\hat{F}} [\hat{F}(x)^2] - \left(\mathbb{E}_{\hat{F}} [\hat{F}(x)] \right)^2. \quad (8)$$

Again, we estimate the covariance given $\{\omega_1, \dots, \omega_n\}$. Since we assume that \tilde{F} is second-order stationary, the covariance of \tilde{F} depends only on the distance between two points, *i.e.*

$$\mathbb{E}_{\tilde{F}} [\tilde{F}(x_k) \tilde{F}(x_{k'})] = \tilde{C}(\|x_k - x_{k'}\|_2) - \mu^2. \quad (9)$$

By setting $\mu = \mathbb{E}_{\tilde{F}} [\tilde{F}(x)]$, we arrive at:

$$V_{\tilde{F}|\omega_1, \dots, \omega_n}(x) = \sum_{(k, k') \in [1, (2 \times \mathcal{K} + 1)^2]^2} \omega_k(x) \omega_{k'}(x) \tilde{C}(\|x_k - x_{k'}\|_2) \quad (10)$$

where the covariance $\tilde{C}(\cdot)$ is estimated from the low resolution image \tilde{f} . The variance depends on the position x . Note, however, since we directly estimate the weights $\omega(x^*)$ without making assumptions on the covariance function, we cannot compute the variance and an approximation is needed to estimate the covariance empirically from \tilde{f} . Eq. 10 however is particularly interesting since it shows that the estimator variance at x is directly proportional to the variance of \tilde{f} and the different values of ω near x . The bigger these values are, the more uncertain the estimator is.

3.4 Network Implementation and Training

⁵ Of course, we cannot account for the low resolution process that produced \tilde{F}

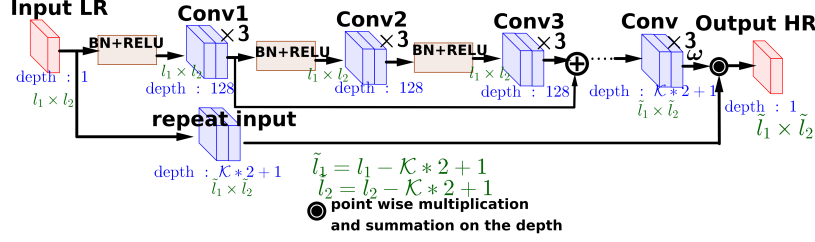


Fig. 1. Overview of the network structure. The input image goes through two branches. The first one calculates the weights ω ; the second copies the input image for an efficient application of the weights via point-wise multiplication.

So far, the theory which we have presented on supervised kriging is generic and does not indicate how one should minimize the risk in Eq. 7 to solve for the kriging weights. We solve for the weights with a CNN network which we show in Fig. 1. The weight estimation branch is composed of a residual network of 9 residual units; the residual unit itself has 2 convolutional layers, each preceded by a batch normalization [12] and a ReLU.

This branch follows a similar architecture as [25, 14], with the difference that they apply this architecture to directly estimate the HR image, while we use it to estimate our kriging weights, *i.e.* as a local dynamic filter [3]. In addition, contrary to [25] our network is not recurrent.

The network learns a total of 20 convolution layers to determine the kriging weights. The first 19 are of depth 128, the last prediction layer has depth $(2 \times \mathcal{K} + 1)^2$ and outputs the kriging weight vector $\omega(x) = [\omega_1(x), \dots, \omega_{(2 \times \mathcal{K} + 1)^2}(x)]$. The weights are applied to the repeated input image via a point-wise multiplication and summation along the depth dimension to yield the HR output. Overall, the network with $\mathcal{K} = 3$ is very lightweight, inference on a 320×480 sized image takes only 0.10 seconds with a titan X.

Note our network is actually learning how to estimate the kriging weights ω , *i.e.* as a local dynamic filter. To derive this dynamic filter end-to-end, we adapt the following formulation of the Eq. 5 in convolution terms:

$$\tilde{f}_i(x^*) = \sum_{k \in \{\|x_k - x^*\|_1 \leq \mathcal{K}\}} \omega_k(x^*) (\tilde{f}_i * h_k)(x_k) \quad \text{with} \quad \sum_{k \in \{\|x_k - x^*\|_1 \leq \mathcal{K}\}} \omega_k(x^*) = 1 \quad (11)$$

with $h_k(x) = \delta(x - x_k)$ or the Dirac function. The h_k filter is denoted by “repeat input” in Fig. 1, since applying the Dirac functions in each position x^* is a way to extract the pixel neighbourhood at position x^* . After having determined the weights, we normalize them so that they sum up to 1, as given by the constraint in Eq. 11. We point-wise multiply the normalized weights to the output of the repeat filter; to arrive at the super-resolved image, we simply sum along the depth.

We implement our network in Tensorflow, minimizing the loss in Eq. 7 with the Adam optimizer, a batch size of 8 and a learning rate of 10^{-4} . We apply

gradient clipping and dropout between the last ReLU and the convolution layer that predicts ω , with a dropout rate of 0.3. The effective depth of our network is $d = 2 + 2 \times U$, with U the number of times the residual units are applied. In our case, we use $U = 9$.

4 Experiments

4.1 Datasets, Pre-processing and Evaluation

For training, we follow [25,14] and use the 291 images combined from [31] and the Berkeley Segmentation datasets [19]. We further augment the data by rotating (90° , 180° and 270°) and scaling the data ($\times 2$, $\times 3$, $\times 4$). We limit to these fixed grades of rotation and scaling to follow the same protocol as [25]. Like [25,14], we work with patches of size 31×31 , sampled from the augmented dataset with a stride of 21. Testing is then performed on the four commonly used benchmarks: *Set 5* [2], *Set 14* [15], *B100* [19] and *Urban 100* [11]. To be consistent with [25,14], we convert the RGB images to YCbCr and only resolved the Y component. This is then combined with an upsampled Cb and Cr component and converted back into RGB. Up-sampling is done via bi-cubic interpolation.

We evaluate the resulting super-resolved images quantitatively with the peak signal-to-noise ratio (PSNR): $\text{PSNR}(\tilde{f}, f_{\text{ref}}) = \log_{10} \left(\frac{255^2}{\sum_{i=1}^N (\tilde{f}(x_i) - f_{\text{ref}}(x_i))^2} \right)$, where f_{ref} and \tilde{f} are the ground truth and the super-resolved images respectively and N refers to the number of pixels in the (high-resolution) image. A higher PSNR corresponds to better results. We also evaluate using Structural Similarity (SSIM) [27], which measures the similarity between two images. The closer the SSIM value is to 1, the more similar the super-resolved image to the ground truth high resolution image.

4.2 Comparison to state-of-the-art

We compare our PSNR and SSIM measures for the different scales against state-of-the-art in Table 1. The first two methods, bi-cubic and local kriging are unsupervised, while all others are supervised approaches. We use the Matlab implementation of bicubic interpolation.

For the local kriging, we use a neighbourhood of 90×90 and a stride of 81. At this sparse setting, unsupervised kriging does very well in comparison to bicubic interpolation. However, it is already extremely slow, since for each patch, we need to compute the empirical covariance, true covariance, and then invert the covariance matrix. In total, it takes approximately 1 minute for an image of size 320×480 . In comparison, our proposed deep kriging, on the same image, applied densely is one magnitude faster, and takes only 0.25 seconds with a titan x.

Looking at our approach (reported in the second last column in Table 1) with respect to the supervised methods, our performance is comparable to the DRRN [25] B1U9 setting, DRCN [15] and VSDR [14]; all three have networks with

depth similar to ours. The best results are reported by the DRRN B1U25 [25], which has 50 convolution layers and is more than twice as deep as our network.

Given the trend of the community to work on deep learning-based approaches for super-resolution, as well as the fact that no labelled data is required for training, one can work with (infinitely) large datasets [26,17]. For fair comparison with state-of-the-art, however, we omit from this table the techniques which do not use the fixed 291 image training set as per the training protocol set by [25,14].

4.3 Model uncertainty

One of the key strengths of our technique is the fact that we can estimate model uncertainty. We show in Figure 2 the estimated variance for each pixel based on Equation 10. To evaluate this equation, we need to apply covariance models for f and \tilde{f} . We do this by first estimating the empirical covariance $c(\tau) = \sum_i f(x_i)f(x_i + \tau) - \mu$ and then solving for C_0 and σ of the Gaussian model in section 3.1 that is the closest to the empirical covariance.

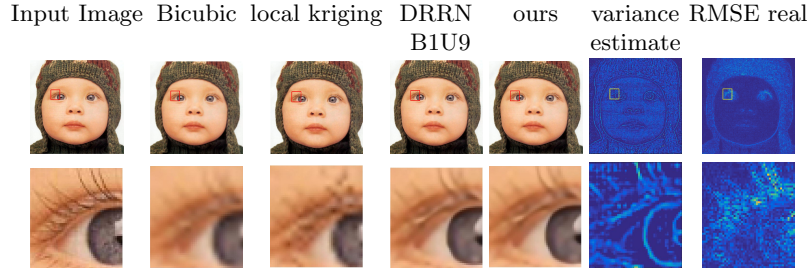
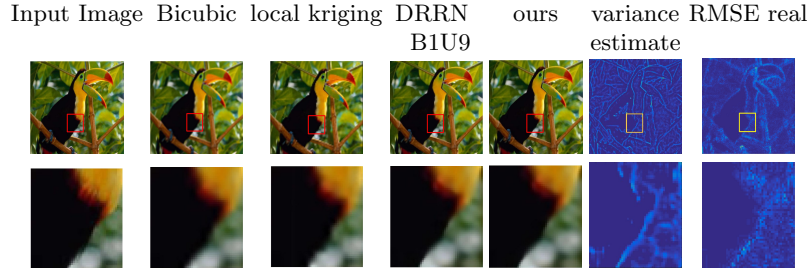
The main advantage of our model of uncertainty is that we can have information on the black box CNN. It gives us feedback about the reliability of results provided by an unknown image. One can see in Figure 2 our uncertainty estimate. The estimated variance has a higher value than the real PSNR this is due to the fact that we have a noisy estimation of the PSNR from a low resolution image. Holistically, however, they are similar, in that areas with high variance corresponds to high PSNR. Quantitatively, we find for Set 5 that 91.9% of the super-resolved pixel values fall within 3 standard deviations based on the estimated variance. In addition, we evaluate the similarity between the images resulting from the PSNR and the one resulting from the variance in set 5. We use the correlation to measure the similarity and find that these images have high correlation up to 0.8.

5 Conclusions

In this paper, we have proposed a joint deep learning and statistical framework for single image super-resolution based on a form of supervised kriging. We solve for the kriging weights in the manner of a local dynamic filter and apply it directly to the low resolution image, all within a single network that can be learned end-to-end. Since we work within the known statistical framework of kriging, we can estimate model uncertainty, something typically not possible for deep networks. More specifically, we show through derivations that the statistical estimator generated by our network is unbiased and we calculate its variance.

References

1. Anbarjafari, G., Demirel, H.: Image super resolution based on interpolation of wavelet domain high frequency subbands and the spatial domain input image. ETRI journal **32**(3), 390–394 (2010)

(a) Super resolution results of set 5 data set with scale $\times 3$.(b) Super resolution results of set 5 data set with scale $\times 3$.**Fig. 2.** Super resolution results

Dataset	Scale	Bicubic	local kriging	SRCNN	VDSR	DRCN	DRRN B1U9	DRRN B1U25	ours
		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
Set 5	$\times 2$	33.66/0.930	35.68/0.923	36.66/0.953	37.53/0.959	37.63/0.959	37.66/0.959	37.74/0.959	37.65/0.960
	$\times 3$	30.39/0.868	31.72/0.864	32.75/0.909	33.66/0.921	33.82/0.923	33.93/0.923	34.04/0.924	33.94/0.923
	$\times 4$	28.42/0.810	29.83/0.814	30.48/0.862	31.35/0.883	31.53/0.885	31.58/0.886	31.68/0.889	31.56/0.886
Set 14	$\times 2$	30.24/0.869	31.26/0.874	32.45/0.907	33.03/0.912	33.04/0.912	33.19/0.913	33.23/0.914	33.10/0.912
	$\times 3$	27.55/0.774	27.94/0.781	29.30/0.822	29.77/0.831	29.76/0.831	29.94/0.831	29.96/0.834	29.81/0.831
	$\times 4$	26.00/0.703	26.43/0.715	27.50/0.751	28.01/0.767	28.02/0.767	28.02/0.767	28.18/0.770	28.08/0.771
B 100	$\times 2$	29.56/0.843	31.06/0.848	31.36/0.888	31.90/0.8960	31.85/0.894	32.01/0.897	32.05/0.897	32.01/0.896
	$\times 3$	27.21/0.8431	27.86/0.7357	31.36/0.8879	28.82/0.7976	28.80/0.7963	28.91/0.7992	28.95/0.800	28.82/0.800
	$\times 4$	25.96/0.668	26.55/0.665	26.90/0.711	27.29/0.725	27.23/0.723	27.35/0.726	27.38/0.728	27.36/0.727
Urban 100	$\times 2$	26.88/0.840	28.32/0.843	29.50/0.895	30.76/0.914	30.75/0.913	31.02/0.916	31.23/0.919	30.95/0.921
	$\times 3$	24.46/0.735	25.06/0.715	29.50/0.799	27.14/0.715	27.15/0.828	27.38/0.833	27.53/0.838	
	$\times 4$	23.14/0.658	23.70/0.627	24.52/0.722	25.19/0.752	25.14/0.751	25.35/0.757	25.44/0.764	25.10/0.749

Table 1. Results of various super resolution techniques. The column 3th and 4th are unsupervised. The other columns are trained on the 291 data set [31,19]. The Urban 100 set having no images at scale 3 we did not evaluate it.

2. Bevilacqua, M., Roumy, A., Guillemot, C., Alberi-Morel, M.: Low-complexity single-image super-resolution based on nonnegative neighbor embedding (2012)
3. Brabandere, B.D., Jia, X., Tuytelaars, T., Gool, L.V.: Dynamic filter networks. In: NIPS (2016)
4. Chang, H., Yeung, D., Xiong, Y.: Super-resolution through neighbor embedding. In: CVPR. vol. 1, pp. I–I. IEEE (2004)
5. Cressie, N.: Statistics for spatial data. John Wiley & Sons (2015)
6. Damianou, A., Lawrence, N.: Deep gaussian processes. In: Artificial Intelligence and Statistics. pp. 207–215 (2013)
7. Dong, W., Zhang, L., Shi, G., Wu, X.: Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. IEEE Transactions on Image Processing (TIP) **20**(7), 1838–1857 (2011)
8. Donoho, D.L.: Compressed sensing. IEEE Trans. on Information Theory **52**(4), 1289–1306 (2006)
9. Freeman, W., Jones, T., Pasztor, E.: Example-based super-resolution. IEEE Computer Graphics and Applications **22**(2), 56–65 (2002)
10. He, H., Siu, W.C.: Single image super-resolution using Gaussian process regression. In: CVPR (2011)
11. Huang, J., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: CVPR. pp. 5197–5206 (2015)
12. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning. pp. 448–456 (2015)
13. Keys, R.: Cubic convolution interpolation for digital image processing. IEEE Trans. on Acoustics, Speech, and Signal Processing **29**(6), 1153–1160 (1981)
14. Kim, J., Lee, L.K., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. In: CVPR (2016)
15. Kim, J., Lee, L.K., Lee, K.M.: Deeply-recursive convolutional network for image super-resolution. In: CVPR (2016)
16. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
17. Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single image super-resolution. In: CVPR Workshops (2017)
18. Marquina, A., Osher, S.: Image super-resolution by tv-regularization and bregman iteration. Journal of Scientific Computing **37**(3), 367–382 (2008)
19. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: ICCV. vol. 2, pp. 416–423 (July 2001)
20. Matheron, G.: Random sets and integral geometry. John Wiley & Sons (1975)
21. Meier, F., Hennig, P., Schaal, S.: Local gaussian regression. arXiv preprint arXiv:1402.0645 (2014)
22. Peleg, T., Elad, M.: A statistical prediction model based on sparse representations for single image super-resolution. IEEE Transactions on Image Processing (TIP) **23**(6), 2569–2582 (2014)
23. Pronzato, L., Rendas, M.J.: Bayesian local kriging. Technometrics pp. 1–12 (2017)
24. Sajjadi, M.S.M., Scholkopf, B., Hirsch, M.: EnhanceNet: Single image super-resolution through automated texture synthesis. In: ICCV (Oct 2017)
25. Tai, Y., Yang, J., Liu, X.: Image super-resolution via deep recursive residual network. In: CVPR (June 2017)
26. Tong, T., Li, G., Liu, X., Gao, Q.: Image super-resolution using dense skip connections. In: ICCV (Oct 2017)

- 27. Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E.: Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing (TIP)* **13**(4), 600–612 (2004)
- 28. Wilson, A.G., Hu, Z., Salakhutdinov, R., Xing, E.P.: Deep kernel learning. In: *AISTATS* (2016)
- 29. Xu, L., Jia, J.: Two-phase kernel estimation for robust motion deblurring. In: *ECCV*. pp. 157–170. Springer (2010)
- 30. Yang, J., Wang, Z., Lin, Z., Cohen, S., Huang, T.: Coupled dictionary training for image super-resolution. *IEEE Transactions on Image Processing (TIP)* **21**(8), 3467–3478 (2012)
- 31. Yang, J., Wright, J., Huang, T., Ma, Y.: Image super-resolution via sparse representation. *IEEE Transactions on Image Processing (TIP)* **19**(11), 2861–2873 (2010)
- 32. Zhang, K., Gao, X., Tao, D., Li, X.: Single image super-resolution with non-local means and steering kernel regression. *IEEE Transactions on Image Processing (TIP)* **21**(11), 4544–4556 (2012)
- 33. Zhao, S., Han, H., Peng, S.: Wavelet-domain hmt-based image super-resolution. In: *International Conference on Image Processing (ICIP)*. vol. 2, pp. II–953. IEEE (2003)