# Image Super-Resolution Using Local Learnable Kernel Regression

Renjie Liao and Zengchang Qin

Intelligent Computing and Machine Learning Lab
School of Automation Science and Electrical Engineering
Beihang University, Beijing, China
lrjconan@gmail.com, zcqin@buaa.edu.cn

**Abstract.** In this paper, we address the problem of learning-based image super-resolution and propose a novel approach called Local Learnable Kernel Regression (LLKR). The proposed model employs a local metric learning method to improve the kernel regression for reconstructing high resolution images. We formulate the learning problem as seeking multiple optimal Mahalanobis metrics to minimize the total kernel regression errors on the training images. Through learning local metrics in the space of low resolution image patches, our method is capable to build a precise data-adaptive kernel regression model in the space of high resolution patches. Since the local metrics split the whole data set into several subspaces and the training process can be executed off-line, our method is very efficient at runtime. We demonstrate that the new developed method is comparable or even outperforms other super-resolution algorithms on benchmark test images. The experimental results also show that our algorithm can still achieve a good performance even with a large magnification factor.

## 1 Introduction

The basic idea of Super-Resolution is to estimate a high resolution (HR) image from a single or several original low resolution (LR) images. It is an inherently ill-posed inverse problem since the mapping between HR image and LR image is many-to-one and much information is lost in the HR-to-LR process. Various methods have been proposed to solve this underdetermined mapping. Roughly, they can be divided into three major categories: (1) Interpolation based methods that generate HR image using single LR image [6,7]. (2) Reconstruction based methods using multiple LR images with some smoothness priors [8,9]. And (3) learning based methods (or example based methods) that use a large training set of HR/LR patch pairs [10,19]. Though the implementation of interpolation based methods are very fast, they are unable to produce sharp edges and clear details. As for conventional reconstruction based methods, since the accurate subpixel motion estimation is extremely difficult [9], their performance is very limited. Such limitations are analyzed using perturbation theory of linear systems by [11]. In learning based methods, Freeman *et al.* [10] learn a Markov Random Field

K.M. Lee et al. (Eds.): ACCV 2012, Part III, LNCS 7726, pp. 349–360, 2013.

(MRF) from large number of generic HR/LR image patch pairs, and infer the HR image patches through belief propagation given LR image patches. They [12] later propose an approximation method for replacing the inference procedure in MRF with a nearest neighbour search framework that speeds up the original belief propagation algorithm and achieves comparable results. Chang *et al.* [13] extend this framework from nearest neighbour to a local regression model based on the idea of Local Linear Embedding (LLE) [15]. This local regression model needs less training patches [10] without losing much expression ability of the whole training database. In recent research, Fattal *et al.* [19] investigate the gradient fields of LR and HR images and devise a new upsampling method. After that, Sun *et al.* [16] uses Gradient Profile Prior (GPP) to model the relationship between sharp and blurred edges, and learn this prior from a training set of natural images. Glasner *et al.* [5] raise the issue of patch redundancy in a single image. They constructed a pyramid framework and utilize other reconstruction based constraints to exploit this phenomenon for single image super-resolution. Inspired by latest progress of compress sensing, Yang *et al.* [3] argue that the sparse representation in the space of LR patches can apply to the space of HR patches. Thus reconstruction of HR image can be solved via sparse coding.

In this paper, our research will focus on learning based methods. To quote experimental results in [5]: "the main improvement in resolution comes from the Example-Based SR component in combined framework". Extensive training may bring substantial benefit to resolution enhancement. Unfortunately, no theories have been given to investigate how many examples are enough to achieve the specific resolution. It seems that is the case of " the more examples, the better results". Typically, millions of HR/LR example patch pairs as required in [17,10] and it is inefficient in both speed and memory. To overcome this main disadvantage and improve the super-resolution quality, we propose a local regression scheme referred to as *Local Learnable Kernel Regression* (LLKR). In this model, specific kernel shapes are learned by metric learning to maintain the expression ability of the dictionary. Meanwhile, since the metric learning is not conducted globally, our model captures more local information, thus making regression more precise and algorithm more efficient.

The rest of the paper is organized as follow: Section 2 introduces the super-resolution problem approached by kernel regression. Section 3 presents the framework of LLKR model and the dictionary construction at length. The effectiveness of our model is verified through a series of experiments on benchmark image sets in Section 4. At last, we concludes the paper and discusses some future work.

## 2 Super-Resolution via Kernel Regression

In this section, we will briefly demonstrate the learning based image super-resolution problem and approach it with classical kernel regression.

Given dictionaries of LR and HR image patches $(\mathbb{D}_l, \mathbb{D}_h)$, for a low resolution image $\mathbf{x}$, estimate its corresponding high resolution image $\mathbf{y}$, subjected to the reconstruction constraint: $\mathbf{x} = D\mathbf{y}$, where $D$ is a downsampling operator. To

solve this inverse problem, we can rely on regression models. Kernel regression is such a non-parametric technique to estimate the conditional expectation of a random variable, among which Nadaraya-Watson kernel regression [1] is most commonly used.

Specifically, training dictionaries $\mathbb{D}_l$, $\mathbb{D}_h$ consists of $n$ pairs of LR (usually represented as image features) and HR patches: $(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), ..., (\mathbf{x}_n, \mathbf{y}_n)$. In order to minimize training error, we estimate an unknown function $f : R^{dx} \rightarrow R^{dy}$ in the form of $\mathbf{y} = f(\mathbf{x}) + \boldsymbol{\varepsilon}$, where $\mathbf{x} \in R^{dx}$, $\mathbf{y} \in R^{dy}$, $\boldsymbol{\varepsilon}$ is the noise, $dx$ and $dy$ are the dimensions of LR patch and HR patch. The Nadaraya-Watson estimator is:

$$\hat{f}(\mathbf{x}) = \frac{\sum_{i=1}^{m} K(\mathbf{x} - \mathbf{x}_i)\, \mathbf{y}_i}{\sum_{i=1}^{m} K(\mathbf{x} - \mathbf{x}_i)} \tag{1}$$

where $K(\cdot) \geq 0$ is a kernel function (e.g. Gaussian, spherical, polynomial and etc). Generally, the mass of the kernel functions mainly lies in the neighbourhood support of $\mathbf{x}$. From the Eq. (1), we can explicitly find that the value of the function $f$ at specific point $\mathbf{x}$ is the locally weighted average of the function values of its neighbouring points. If we use kernel density representation to approximate the joint probability $P(\mathbf{x}, \mathbf{y})$ and marginal probability $P(\mathbf{x})$, then the conditional probability can be expressed as:

$$P(\mathbf{y}_i \,|\mathbf{x}) = \frac{K(\mathbf{x} - \mathbf{x}_i)}{\sum_{i=1}^{m} K(\mathbf{x} - \mathbf{x}_i)} \tag{2}$$

Thus the Nadaraya-Watson kernel regression is actually the conditional expectation $E[\mathbf{y} \,|\mathbf{x}]$.

## 3 Local Learnable Kernel Regression Model

Before introducing our model, we first investigate how to make kernel regression adaptive through metric learning which escapes tuning parameters of kernel.

### 3.1 Metric Learning for Kernel Regression

Since the kernel functions in Eq. (2) determines the conditional probability, choosing the forms of kernel function and fitting the kernel parameters are the key issues for the performance of kernel regression. Weinberger [2] proposed the metric learning for kernel regression (MLKR) model to learn an appropriate distance function of kernel. The Mahalanobis metric is used to parameterize the kernel function (1):

$$K(\mathbf{x} - \mathbf{x}_i) = \exp\left\{-(\mathbf{x} - \mathbf{x}_i)^T \cdot M \cdot (\mathbf{x} - \mathbf{x}_i)\right\} \tag{3}$$

where $M$ is a symmetric positive semi-definite real matrix (Euclidean metric is a special case of Mahalanobis metric if setting $M$ to identity matrix). Therefore,

we are able to calculate the regression value via Eq. (1). It is straightforward to define the loss function $E$ as the summed squared error $E$ given below.

$$E = \sum_{i=1}^{n} \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|_2^2 \tag{4}$$

where $\hat{\mathbf{y}}_i = \hat{f}(\mathbf{x}_i)$. With this objective function, the metric learning problem is formulated as constrained optimization:

$$\min_{M} \; E, \qquad s.t.\; M \geq 0 \tag{5}$$

However, learning matrix $M$ directly requires enforcing a positive semi-definite constraint during optimization which is non-linear and expensive to satisfy. To learn $M$ cheaply, we can decompose it by: $M = A^T A$. If we substitute $M$ in equation (3) with this factorization in terms of $A$, the original Mahalanobis metric is equivalent to Euclidean metric on the data points after applying linear transformation of $\mathbf{x} \rightarrow A\mathbf{x}$. Since $A$ is an unconstrained real matrix, the hard constraint prior is eliminated. Thus we can now cast the metric learning as an unconstrained matrix optimization problem $\min_{A} \; E$. Note that one can adjust the number of parameters according to the complexity of the regression task through setting $A$ to square, triangular or diagonal matrix.

### 3.2 Local Metric vs Global Metric

Though MLKR provides an effective way to improve kernel regression, it suffers drawbacks for applying directly to super-resolution. The premise for the superior performance of MLKR is that, most local neighbourhood structures of data could be shaped properly under a single global linear transformation. However, this assumption is very likely to be violated in super-resolution, since downsampling process from HR to LR patches is a many-to-one mapping and collapses the local neighbourhood structures of HR patches which makes LR patches distributed like multi-modal. Thus, reconstructing HR patch through global metrics learnt from LR patch is problematic. We therefore seek the potential of replacing it with multiple local linear transformation (i.e local metrics).

To analyze efficiencies of multiple local metrics and single global metric in the context of super-resolution, we first conduct an illustrative experiment. In the stage of collecting data, we randomly sample about 100,000 HR patches with size 5 from *Berkeley Segmentation Database*. And we obtain corresponding LR patches through downsampling operator with factor 3.0. To construct dictionaries, we compute gradients as the representation for LR patches like in [3]. After these steps, we conduct MLKR with the whole data set to learn one global metric and record the minimum value of total squared regression error $E$. Then we repeatedly perform k-means algorithm for clustering with different number of clusters. More specifically, every time we obtain one partition of the data set, we implement MLKR inside each cluster and record the minimum value of $E$. The step length for number of clusters is set to 10 and we plot the curve of $E$ varied

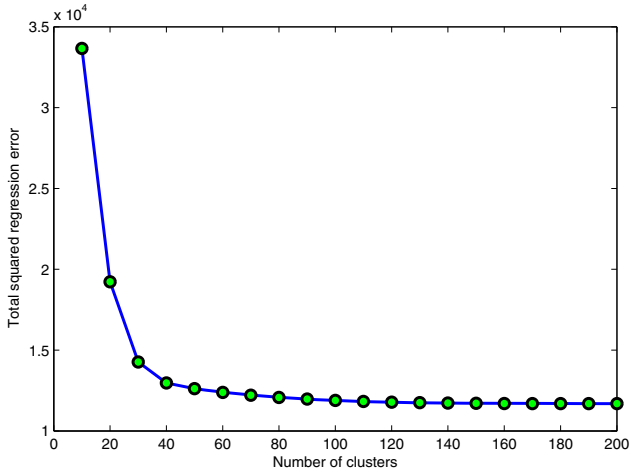



**Fig. 1.** Total squared regression error $E$ varies with the number of clusters

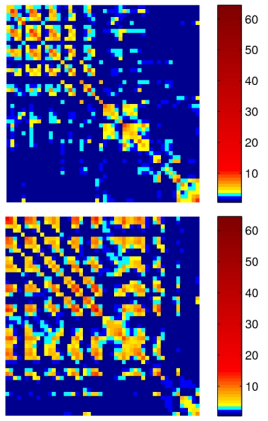



**Fig. 2.** Two examples of learned local metric matrices

with number of clusters in Fig. 1. From this figure, we can find that total squared regression error $E$ decreases a lot with number of clusters increase to 40 and then it tends to be stable. Therefore, for the dictionary used in super-resolution, local metric learning for kernel regression is more beneficial than MLKR which learns only one global metric for all data points.

### 3.3 Local Metric Learning

Inspired by the previous experiment, we thus naturally extend MLKR to learn multiple local Mahalanobis metrics in LR patch space. The approach is summarized in Algorithm 1. This algorithm is referred to as *Local Learnable Kernel Regression* (LLKR) for the reason that specific forms of metrics are learnt through minimizing regression errors in corresponding local spaces of LR patches. Therefore, we need to solve a MLKR problem inside each cluster. The gradient of the objective function is derived as:

$$\frac{\partial E}{\partial A} = 4A \sum_{k=1}^{dy} \sum_{i=1}^{n} (\mathbf{y}_{ik} - \hat{\mathbf{y}}_{ik}) \sum_{j \in \Omega_i} \left(\mathbf{y}_{jk} - \hat{\mathbf{y}}_{ik}\right) \tilde{K}_{ij} \mathbf{d}_{ij} \mathbf{d}_{ij}^T \tag{6}$$

where $\mathbf{d}_{ij} = \mathbf{x}_i - \mathbf{x}_j$ and $\tilde{K}_{ij} = K_{ij} \Big/ \sum_{j \in \Omega_i} K_{ij}$. For the metric initialization, $A$ is set to an identity matrix with some random disturbance on the diagonal elements. This optimization problem can be solved efficiently through regular gradient based optimization algorithms such as conjugate gradient, BFGS or stochastic gradient descent. Comparing to MLKR model, our method has one additional parameter - the number of clusters, which is convenient for incorporating prior knowledge. Roughly speaking, if the magnification factor is large

**Algorithm 1.** (Local Learnable Kernel Regression)

**Input:** training dictionaries ($\mathbb{D}_l$, $\mathbb{D}_h$), number of clusters $p$

1: Find $p$ clusters of dictionary $\mathbb{D}_l$ using k-means.
2: **for** $i = 1$ to $p$ **do**
3: Specify cluster assignment set $C_i$ for $i$th cluster.
4: Specify neighbourhood index set $\Omega_j$ for $\mathbf{x}_j$, $j \in C_i$.
5: solve

$$\min_{A_i} \sum_{j \in C_i} \left\| \mathbf{y}_j - \hat{\mathbf{y}}_j \right\|_2^2$$

$$s.t.\ \hat{\mathbf{y}}_j = \left( \sum_{m \in \Omega_j} K_{jm} \cdot \mathbf{y}_m \right) \Bigg/ \left( \sum_{m \in \Omega_j} K_{jm} \right)$$

$$K_{jm} = \exp\left\{ -(\mathbf{x}_j - \mathbf{x}_m)^T {A_i}^T A_i (\mathbf{x}_j - \mathbf{x}_m) \right\}$$

6: **end**

**Output:** local metrics $\{A_1, ..., A_p\}$. Cluster centers of dictionary $\mathbb{D}_l$: $\{x_{C_1}, ..., x_{C_p}\}$.

which means the collapsing effect fore-mentioned is obvious, the number of clusters should be increased and vice versa. Through experiments we found that, on average, setting number of clusters to 100 is best for dictionary with around 100,000 patches and using more clusters will take the risk of over-fitting. Furthermore, since the complexity of MLKR is approximately $O\left(n^2\right)$, it is prohibitive to apply when the number of samples is very large (e.g. when the size of the dictionary used for super-resolution is approaching to 1 million). Fortunately, the algorithmic complexity of LLKR is $O\left(n^2/p\right)$ with balanced clustering - when the numbers of data points in each cluster are almost the same. Two local metrics learnt in our experiments are visualized in Fig. 1 and 2. The dimension of each matrix is $50 \times 50$.

### 3.4 Local Metrics for Super-Resolution

In this section, we discuss how to exploit learnt local matrices from training stage for constructing super-resolution images. Given a LR patch $\mathbf{x}$, we first calculate which cluster it belongs to. This step is always very efficient since the number of clusters is much less than the size of the dictionary. Next, within this cluster, we apply regression formulas of Eq. (1) and (3) based on the corresponding learnt metric and reconstruct HR patch $\mathbf{y}$. As we have discussed in pervious section, we are only interested in local neighbouring points for kernel regression, we conduct the nearest neighbours search and set kernel value to zero if it is less than a predefined threshold (e.g. $e^{-10}$). Relying on high dimensional data structure like kd-trees [4], the neighbourhood seeking step can be speed up significantly. Moreover, for each cluster, the kd-tree building process can be done off-line. It is pretty obvious that, comparing to the whole dictionary, the number of patches inside each cluster is modestly small, thus making our algorithm very fast. The pseudo-code for super-resolution image construction is shown in Algorithm 2.

**Algorithm 2.** (Super-resolution with LLKR)

**Input:** training dictionaries $(\mathbb{D}_l, \mathbb{D}_h)$, local metrics $\{A_1, ..., A_p\}$, cluster centers of dictionary $\mathbb{D}_l$ $\{x_{C_1}, ..., x_{C_p}\}$, low resolution image $\mathbf{X}$.

1: **for** every patch $\mathbf{x} \in \mathbf{X}$ crawled in raster scan from upper-left to bottom-right with fixed step **do**

2: Find the nearest neighbour cluster label $i$ according to: $\min_i \ \|\mathbf{x} - \mathbf{x}_{C_i}\|_2^2$

3: Specify the local neighbourhood $\Omega$ for $\mathbf{x}$ in the $i$th cluster.

4: Reconstruct the high resolution patch $\mathbf{y}$ as follow:

$$K_j = \exp\left\{-(x - x_j)^T {A_i}^T A_i (x - x_j)\right\} \quad j \in \Omega$$
$$y = \left(\sum_{j \in \Omega} K_j \cdot y_j\right) \Big/ \left(\sum_{j \in \Omega} K_j\right)$$

5: **end**

6: Refine $\mathbf{Y}$ with the reconstruction constraint through back-projection.

**Output:** high resolution image $\mathbf{Y}$.

### 3.5 Dictionary Construction

Dictionary plays an important role in learning based super-resolution algorithms [10,17]. Usually, enormous dictionaries are required for the learning process. Here we propose two preprocessing methods to help reducing the size of dictionary.

**Random Sampling with Gradient Prune.** The basic differences between HR images and LR images are the high frequency information. They often occur in edges and corners of the HR image [10]. To investigate this phenomenon, we adopt a random sampling method with gradient prune strategy. When a random raw HR image patch is sampled, we calculate the magnitude of its gradient and then discard this patch if this value is less than a predefined threshold (e.g. 20). By doing so, we can reduce the number of patches, which are redundant in the dictionaries and thus not helpful in improving regression performance.

**Feature Extraction and Contrast Normalization.** In order to construct a dictionary $\mathbb{D}_l$ with rich expression ability, we should keep one point in mind, the LR image patch should be as informative as possible. High-pass filter is adopted by Freeman *et al.* [10] to obtain such patch. Baker *et al.* [18], Chang *et al.* [13] and Yang *et al.* [3] all use gradients with different order of gradient to represent the LR patch. Considering both the speed and performance, we only take the normalized 1st order gradient that can be calculated by convolving with Sobel mask. Thus feature of LR patch in $\mathbb{D}_l$ is:

$$\mathbf{x} = \begin{bmatrix} vec\,(G_h) \\ vec\,(G_v) \end{bmatrix} \Big/ \left\| \begin{bmatrix} vec\,(G_h) \\ vec\,(G_v) \end{bmatrix} \right\|_2 \tag{7}$$

where the $vec\,(\cdot)$ is the vectorization operator and $G_h$ $G_v$ are gradient matrices of LR image patch along horizontal, vertical direction respectively. To enlarge the

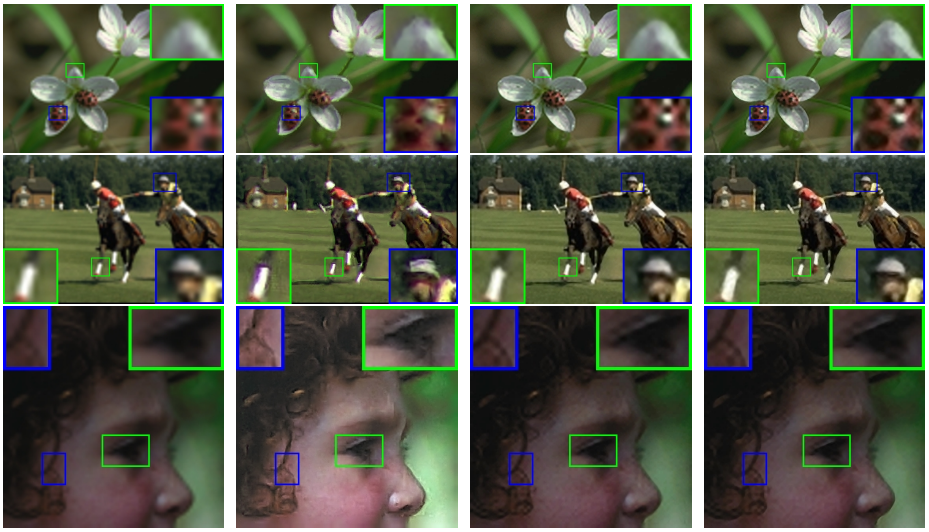
**Fig. 3.** Some of training images used in our experiments and the corresponding visualization of the contrast normalization

generalization ability, Freeman *et al.* [10] take a complex contrast normalization procedure to generate the dictionary $\mathbb{D}_h$. Here we set up a succinct alternative method which is based on the assumption that, the mean value of the HR image patch $\mathbf{y}$ and LR image feature $\mathbf{x}$ is almost the same if the magnification factor is modest. We first subtract the HR image patches with its mean value $\bar{\mathbf{y}}$ and then take the $L2$ normalization to the remainder patch $\mathbf{y} - \bar{\mathbf{y}}$ before saving it. That means, in dictionary $\mathbb{D}_h$, a HR patch $\tilde{\mathbf{y}}$ is:

$$\tilde{\mathbf{y}} = \frac{\mathbf{y} - \bar{\mathbf{y}}}{\|\mathbf{y} - \bar{\mathbf{y}}\|_2} \tag{8}$$

## 4 Experiments

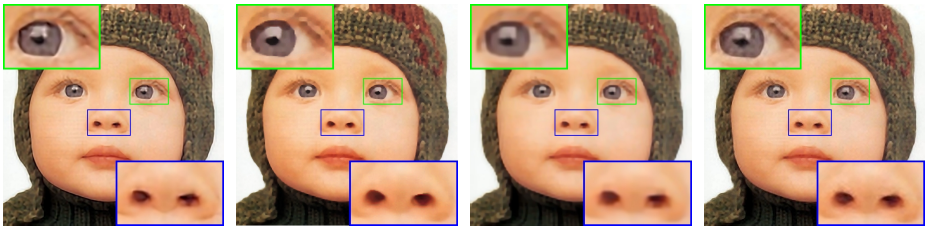
**Experiment Implementations:** Nearly 100,000 patches are collected from 40 images which are downloaded form Flicker to build dictionaries. Some of the training images and the corresponding visualization of contrast normalized ones are shown in Fig. 3. And then we resort to cross-validation on this training dictionary to set proper number of clusterings in LR patches for the initialization of k-means. Within each group, we learn a full square metric matrix. If speed is a concern, the diagonal metric matrix is a good alternative. As mentioned in [2], considering for faster learning, one can cache the nearest neighbours and update them according to the metric every 15 gradient steps. To avoid local minima, random initializations are taken and the outcome with least square error in (3.1) is chosen. In the stage of testing, we apply our algorithm to a serial of images under different magnification factors from 2.0 to 8.0. As a conventional scheme, we take the super-resolution algorithm of magnification factor 2.0 as the base-method. And then we achieve results of greater magnification factor through applying this base-method recursively. Considering the reconstruction constraint as forementioned, we use back-projection to restrain the final output. Through our experiments, we find that this procedure typically converges after 5 to 10 iterations. Another important issue is selecting the size of image patch. Intuitively, the larger the patch, the worse the generalization ability of dictionaries and the slower the speed of learning based algorithm. However, if the patch is too small to capture the width of obvious edges in images, the resolution is hardly to be enhanced. In our experiments, we test different sizes and find that $5 \times 5$ image patch with 1 pixel step size gives the best performance. For



(*a*) Bicubic (*b*) MRF (*c*) Sparse coding (*d*) Our results

**Fig. 4.** Comparison of super-resolution under the magnification factor 4.0, results from Bicubic interpolation, MRF super-resolution [10], super-resolution via Sparse Representation [3] and our method

processing color images, we first convert them from RGB color space to YCbCr and then apply our method only to the Y channel since illuminance changes are most discernable for human vision. As for channels of Cb and Cr, we just magnify them with bicubic interpolation. Lastly, we combine these three channels to obtain the final high resolution image.



(*a*) Path redundancy (*b*) Imposed edge statistics (*c*) Fast upsampling (*d*) Our method

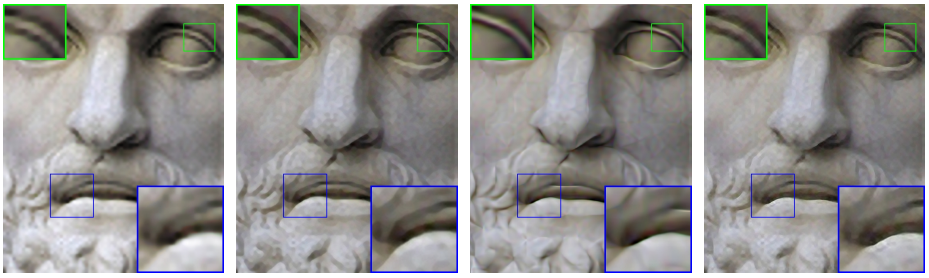
**Fig. 5.** Super-resolution (4×), results using single image super-resolution via patch redundancy [5], imposed edge statistics [19], fast image upsampling [21] and our method

**Experiment Results:** We first compare our method with several learning based super-resolution algorithms. In Fig. 4, comparison of our method with bicubic interpolation, Markov Random Field (MRF) for super-resolution [10],

super-resolution via Sparse Representation [3] are displayed. All images are magnified by factor 4.0. From these figures, it is obvious that images produced by MRF have some synthesized details which make images somehow unreal. And the results of our method are more sharp in edges and less blurred overall than the ones of sparse representation, for example, in the regions of the children's eye and hair, near edges of the petal and the horseshoe, and so on.

**Table 1.** The RMSE, PSNR and SSIM values of different approaches for the Infant image

| Method | RMSE | PSNR | SSIM |
|---|---|---|---|
| Fattal *et al.* [19] | 17.9325 | 23.0580 | 0.5829 |
| Glasner *et al.* [5] | 16.1990 | 23.9410 | 0.5930 |
| Shan *et al.* [21] | 12.3307 | 26.3110 | 0.6494 |
| Our method | **11.9409** | **26.5900** | **0.6971** |



(*a*) Imposed edge statistics (*b*) GPP (*c*) Local self-examples (*d*) Our method

**Fig. 6.** Super-resolution (8×), results using imposed edge statistics [19], GPP [16], local self-examples [20] and our method

To check the performances of our method further, we then conduct comparative experiments with imposed edge statistics [19], single image super-resolution using redundant patches [5], super-resolution using Gradient Profile Prior (GPP) [16], image upscaling from local self-examples [20] and fast image upsampling [21]. Firstly, we investigate the 4× results on the infant picture in Fig. 5. It is clear that the result of fast image upsampling is the most blurred one, and imposed edge statistics generates over-smooth edges with details lost to some extent. Though the image obtained from single image super-resolution is comparatively similar with ours on the whole, we still can find that our method outperforms it since the irregular contours exist with their method in some regions (e.g. the eye ball and the nostril of infant in the zooming up rectangle). To assess these results quantitatively, we evaluate the Root Mean Squared Error (RMSE), Peak Signal to Noise Ratio (PSNR) and Structural Similarity (SSIM)

[14] index for above methods[1]. The measurement values are listed in Table 1. And our method achieves the best results in terms of all these measurements. Then we increase the magnification factor to 8.0 for challenging test. As shown in Figure 6, the result of imposed edge statistics degenerates and is blurred overall in this situation. Compared with our method, GPP tends to produce smoothness along edges which makes the edges wider than original and loses some details. As for the result of local self-examples, it is interesting to note that the edges are more sharp than any other methods. However, since their algorithm is for the purpose of upscaling, it may not obey the reconstruction constraint in the context of super-resolution. Thus the whole image is over-smoothed and renders like artistic style graphic which produces some unreal edges (e.g. the corner of mouth in Figure 6). More high resolution (4×) results of our method compared with bicubic interpolation are displayed in Figure 7.

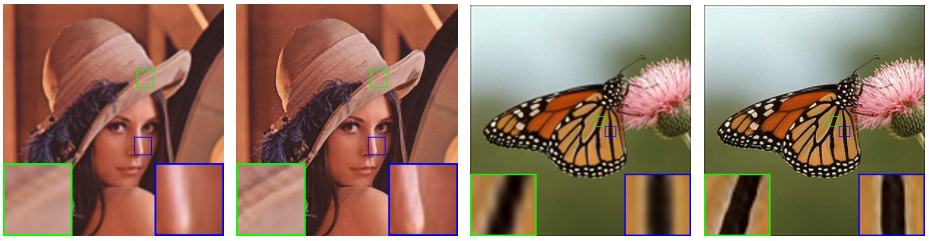


**Fig. 7.** More super-resolution (4×) results. Left images are results of Bicubic interpolation and right are ours.

## 5 Conclusions and Future Work

In this paper we propose a local learnable kernel regression (LLKR) model for learning based image super-resolution. Data-adaptive regression kernels are learnt through metric learning in local regions of LR patches. Each learnt metric captures more information inside the corresponding local region than a single distance metric defined on global space, thus improving the regression precision totally. Our model can be extended to various applications, such as image restoration, reconstruction, and some general regression problems. Future work involves incorporating some priors to automatically choose the number of local metrics and improving the performance through learning distance metrics of other forms.

**Acknowledgement.** This work is partially funded by the NCET Program of MOE, the SRF for ROCS and the Fundamental Research Funds for the Central Universities in China.

[1] The ground truth of the infant image can be obtained from `http://www.cs.huji.ac.il/ giladfreedmn/projects/lss_upscale/supplemental/index.html`

## References


1. Nadaraya, E.A.: On Estimating Regression. Theory of Probability and its Applications 9, 234–778 (1964)
2. Weinberger, K.Q., Tesauro, G.: Metric Learning for Kernel Regression. JMLR 2, 612–619 (2007)
3. Yang, J.C., Wright, J., Huang, T.S., Ma, Y.: Image Super-Resolution Via Sparse Representation. IEEE Trans. Image Process. 19, 2861–2873 (2010)
4. Arya, S., Mount, D.M.: Approximate Nearest Neighbor Queries in Fixed Dimensions. In: SODA, pp. 271–280 (1993)
5. Glasner, D., Bagon, S., Irani, M.: Super-Resolution from a Single Image. In: ICCV (2009)
6. Hou, H.S., Andrews, H.C.: Cubic Splines for Image Interpolation and Digital Filtering. IEEE Trans. Acoustics, Speech, and Signal Processing 26, 508–517 (1978)
7. Li, X., Orchard, M.T.: New Edge Directed Interpolation. In: ICIP (2000)
8. Irani, M., Peleg, S.: Motion Analysis for Image Enhancement: Resolution, Occlusion, and Transparency. J. Visual Communication and Image Representation 4, 324–335 (1993)
9. Takeda, H., Milanfar, P., Protter, M., Elad, M.: Super-Resolution Without Explicit Subpixel Motion Estimation. IEEE Trans. Image Process. 18, 1958–1975 (2009)
10. Freeman, W.T., Pasztor, E.C., Carmichael, O.T.: Learning Low-Level Vision. Int'l J. Computer Vision 40, 25–47 (2000)
11. Lin, Z.C., Shum, H.Y.: Fundamental Limits of Reconstruction-Based Superresolution Algorithms under Local Translation. IEEE Trans. Pattern Anal. Mach. Intell. 26, 83–97 (2004)
12. Freeman, W.T., Jones, T.R., Pasztor, E.C.: Example-Based Super-Resolution. IEEE Computer Graphics and Applications 22, 56–65 (2002)
13. Chang, H., Yeung, D.Y., Xiong, Y.M.: Super-Resolution through Neighbor Embedding. In: CVPR, pp. 275–282 (2004)
14. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. IEEE Trans. Image Process. 13, 600–612 (2004)
15. Roweis, S.T., Saul, L.K.: Nonlinear Dimensionality Reduction by Locally Linear Embedding. Science 290, 2323–2326 (2000)
16. Sun, J., Xu, Z.B., Shum, H.Y.: Image super-resolution using gradient profile prior. In: CVPR, pp. 1–8 (2008)
17. Sun, J., Zheng, N.N., Tao, H., Shum, H.Y.: Image Hallucination with Primal Sketch Priors. In: CVPR, pp. 729–736 (2003)
18. Baker, S., Kanade, T.: Limits on Super-Resolution and How to Break Them. IEEE Trans. Pattern Anal. Mach. Intell., 1167–1183 (2002)
19. Fattal, R.: Image upsampling via imposed edge statistics. ACM Trans. Graph. 26 (2007)
20. Freedman, G., Fattal, R.: Image and video upscaling from local self-examples. ACM Trans. Graph. 30 (2011)
21. Shan, Q., Li, Z.R., Jia, J.Y., Tang, C.K.: Fast image/video upsampling. ACM Trans. Graph. 27 (2008)