

# Learning Hierarchical Decision Trees for Single-Image Super-Resolution

Jun-Jie Huang and Wan-Chi Siu, *Life-Fellow, IEEE*

**Abstract**—Sparse representation has been extensively studied for image super-resolution (SR), and it achieved great improvement. Deep-learning-based SR methods have also emerged in the literature to pursue better SR results. In this paper, we propose to use a set of decision tree strategies for fast and high-quality image SR. Our proposed SR using decision tree (SRDT) method takes the divide-and-conquer strategy, which performs a few simple binary tests to classify an input low-resolution (LR) patch into one of the leaf nodes and directly multiplies this LR patch with the regression model at that leaf node for regression. Both the classification process and the regression process take an extremely small amount of computation. To further boost the SR results, we introduce a SR using hierarchical decision trees (SRHDT) method, which cascades multiple layers of decision trees for SR and progressively refines the estimated high-resolution image. Inspired by the random forests approach, which combines regression models from an ensemble of decision trees, we propose to fuse regression models from relevant leaf nodes within the same decision tree to form a more robust approach. The SRHDT method with fused regression model (SRHDT\_f) improves further the SRHDT method by 0.1-dB in PSNR. Our experimental results show that our initial approach, the SRDT method, achieves SR results comparable to those of the sparse-representation-based method and the deep-learning-based method, but our method is much faster. Furthermore, our enhanced version, the SRHDT\_f method, achieves more than 0.3-dB higher PSNR than that of the A+ method, which is the state-of-the-art method in SR.

**Index Terms**—Classification, decision tree, image processing, regression and training, single-image super-resolution (SR).

## I. INTRODUCTION

**S**INGLE-IMAGE super-resolution (SR) aims to increase the resolution of a single-input low-resolution (LR) image by upsampling, deblurring, and denoising, while the resultant high-resolution (HR) image should preserve the characteristics of natural image, such as sharp edges and rich texture. The LR image  $\mathbf{X}$  in the SR problem is assumed to be a blurred and downsampled version of the original HR image  $\mathbf{Y}$

$$\mathbf{X} = \mathbf{D}\mathbf{H}\mathbf{Y} + \mathbf{n} \quad (1)$$

Manuscript received July 2, 2015; revised October 6, 2015; accepted December 4, 2015. Date of publication December 30, 2015; date of current version May 3, 2017. This work was supported in part by the Center for Signal Processing, The Hong Kong Polytechnic University, and in part by the Research Grants Council through the Hong Kong Government under Grant PolyU5243/13E(B-Q38S). This paper was recommended by Associate Editor X. Li.

The authors are with the Centre for Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong (e-mail: jj.huang@connect.polyu.hk; enwcsiu@polyu.edu.hk).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCSVT.2015.2513661

where  $\mathbf{D}$  is the downsampling operator,  $\mathbf{H}$  is the blur operator, and  $\mathbf{n}$  is the additive noise.

SR has practical significance to break the inherent LR imaging of the devices, such as surveillance systems, satellite imaging, and medical imaging. It can also be applied to image/video coding and compression. The solutions of the super-resolved HR pixels are not unique. A pixel in the input LR image is related to multiple pixels in the resultant HR image depending on the upscaling factor. Thus, SR is an ill-posed underdetermined inverse problem. To mitigate this ill-posed problem, many priors have been adopted to constrain the solution. The existing SR methods in the literature can be classified into three main classes: 1) interpolation-based methods [1]–[9]; 2) reconstruction-based methods [10]–[18]; and 3) learning-based methods [19]–[48].

Interpolation-based methods [1]–[9] use a nonadaptive filter [1]–[4] or adaptive filters [5]–[9] to estimate unknown HR pixels. The nonadaptive methods (e.g., bicubic and bilinear) are based on smoothness prior, and the adaptive methods are mainly based on the geometric duality assumption. The nonadaptive interpolation-based methods are widely used in practical applications due to their low computational complexity; however, their reconstructed HR images are usually with blurry edges and ringing artifacts.

Reconstruction-based methods [10]–[18] impose certain prior knowledge to regularize this ill-posed problem so as to suppress artifacts and reach a solution that is more likely to be the natural image. Some of the commonly used natural image priors include total-variation prior [10], gradient-profile prior [11]–[14], and nonlocal similarity [15]–[18]. Although the reconstruction-based methods can generate sharp edges, the details of the HR image cannot be restored, especially for cases with large upscaling factors.

Learning-based approaches [19]–[48] divide the input LR image into overlapped LR patches and estimate their corresponding HR patches from a set of LR–HR exemplar patch pairs that are cropped from external datasets or the input LR image itself. The essence of learning-based method is to learn an effective and efficient LR–HR patch co-occurrence model for HR patch prediction. In the pioneering work, entitled example-based SR, Freeman *et al.* [19], [20] proposed to search  $k$ -nearest neighbor ( $k$ -NN) LR–HR patch pairs of the input LR patch from an external dataset and estimate the desired HR patch with the retrieved HR patches in a Markov random field framework. Although the super-resolved HR image is much sharper than the bicubic interpolated image, the result appears to be noisy. Chang *et al.* [21]

proposed an SR method based on neighbor embedding (NE), which assumes that the LR patches and their corresponding HR patches share a similar low-dimensional nonlinear manifold. The NE-based approaches [21]–[26] have the same spirit as Freeman’s method [19]; however, the predicted HR patch is reconstructed by its weighted  $k$ -NN, where the weights and retrieved  $k$ -NN are estimated by the input LR patch with LR–HR exemplar patch pairs. The performances of the  $k$ -NN-based approaches [19], [20] and NE-based approaches [21]–[26] are closely related to the number of the LR–HR exemplar patch pairs. While a huge LR–HR patch pair dataset requires vast memory for storage, it could also make the  $k$ -NN searching time become enormous. For an efficient SR, a compact representation of the huge number of LR–HR patch pairs turns necessary. The dictionary-based methods [27]–[44] explicitly or implicitly form a dictionary or coupled dictionaries to represent the relationship between LR and HR patches in the training dataset. The sparse coding SR (ScSR) method proposed in [27] jointly learns two coupled overcomplete dictionaries based on sparse representation for SR. In the sparse coding (SC) framework, each input LR patch can be sparsely represented by a few atoms in the LR dictionary and the desired HR patch is then reconstructed using the same sparse signal with the coupled HR dictionary. Although the NP-hard  $l^0$ -norm problem in the SC framework has been relaxed to an  $l^1$ -norm problem, it is still time consuming to produce the results. Many works [28]–[33] have been proposed to improve the ScSR method in recent years. Especially, Zeyde *et al.* [31] improved the ScSR method in both runtime efficiency and SR quality by reducing the dimensionality of the feature vectors using the principal component analysis (PCA) and applying the orthogonal matching pursuit (OMP) for SC. Peleg and Elad [33] suggested clustering the data and cascading several levels of their proposed statistical prediction model, which estimates the HR patch sparse representation from the LR patch sparse representation and LR sparsity pattern. This simple feedforward neural network-like inference scheme leads to a low-complexity SR algorithm.

To further improve the efficiency of the learning-based single-image SR, recently some fast SR algorithms were proposed in [35]–[44]. Yang and Yang [35] proposed the simple function method, which applies a  $K$ -means clustering method to the training LR patches and uses linear regression to learn a regression model for each cluster. During runtime, the input LR patch is first classified into one of the learned cluster centers and then the regression model belonging to that cluster center is applied for SR. Timofte *et al.* [36] proposed the anchored neighbor regression (ANR) method, which further relaxes the SC problem from  $l^1$ -norm to  $l^2$ -norm. This arrangement has a closed-form solution learned using the neighborhood dictionary atoms. By dividing the dictionary atoms into  $K$  groups, the ANR method learns  $K$  regression models offline. In the SR phase, each input LR patch feature finds its nearest neighbor dictionary atom by solving the SC problem and then uses the corresponding regression model for SR. The same research group further improved the ANR method and proposed the adjusted ANR (A+) method [37], which is the state-of-the-art for fast single image SR, and

achieved the best results in the literature. Instead of using neighborhood dictionary atoms to learn the regression model, the A+ method directly utilizes the dense neighborhood training LR–HR patch pairs of a dictionary atom for learning the regression model, which offers higher accuracy with the same processing time as the ANR method. Dong *et al.* [41] proposed an SR convolutional neural network (SRCNN) model for single-image SR. The SRCNN model consists of three layers of convolutional neural network for patch extraction and representation, nonlinear mapping, and HR image reconstruction, respectively. The SRCNN method offline learns the convolutional neural network using a huge number of training LR–HR patch pairs with millions of backpropagations and applies the learned SRCNN model during testing.

Most of the emerged fast learning-based SR approaches (e.g., simple function method, ANR method, and A+ method) classify the training data into a small number of groups and learn a regression model for each group. However, the classification time of these methods is linearly related to the number of clusters. In the early version of this paper, Huang and Siu [43] proposed to perform image SR using random forests (SRRF) method, which uses random forests to super-resolve the LR image. A similar method [44] that also applies random forests for SR has appeared recently. The random forests approach is an ensemble of decision trees. Each decision tree in the random forests classifies the input LR patch into one of the leaf nodes. The retrieved regression models from the reached leaf nodes are combined to form a more robust regression model for SR. The classification time of a decision tree has a linear relationship with its depth that is approximately the logarithm of the number of leaf nodes. This great property of the decision tree makes it favorable for fast learning-based SR.

In this paper, we propose to use decision tree and hierarchical decision trees for fast and high-quality single-image SR. The contributions of this paper can be summarized into four aspects.

- 1) We demonstrate that decision tree is suitable and capable of fast SR, our proposed SR using decision trees (SRDT) method achieves SR quality comparable with those of Peleg’s method [33] and the SRCNN method [41], while it takes less than 1% running time compared with the SRCNN method or on our further analysis on the computational complexity, our method is only around 6% as that of the SRCNN method.
  - 2) We propose to fuse regression models within a single decision tree to improve the SR performance.
  - 3) We propose to use a hierarchical decision trees framework to further improve the SR quality, and this proposed SR using hierarchical decision trees (SRHDT) method can provide more than 0.3-dB gain in peak SNR (PSNR) compared with the state-of-the-art SR algorithm: the A+ method.
  - 4) We show a sample application for video SR using our proposed SRDT method with the data-dependent model.
- The rest of this paper is organized as follows. Section II introduces the proposed SRDT method and the regression model fusion idea. Section III presents the proposed SRHDT

method. Section IV demonstrates the data-dependent model for image SR using the proposed SRDT method. Section V presents and analyzes the results of our extensive experimental work, and Section VI draws conclusions.

## II. IMAGE SUPER-RESOLUTION USING DECISION TREE

In this paper, we propose to use decision tree and its variations for SR. Decision tree was first proposed by Breiman *et al.* [49] and is now a commonly used data mining algorithm. A decision tree is in a tree structure where a node with two child nodes is called a nonleaf node and a node without a child node is called a leaf node. The nonleaf node is responsible for classification by partitioning the training or the testing data into its left or right child node according to the result of the split function with the learned binary test parameters. At each leaf node, a prediction model is learned using the arrived training data. The testing data are mapped to its desired form with the prediction models. The number of leaf nodes is exponential to the depth of the decision tree, and the classification time is in linear relationship to the depth of the decision tree. With this good property, the decision tree is better than the  $K$ -means clustering and using the SC to perform classification for SR. The tradeoff between image SR quality and the number of regression models can be relieved.

The general idea of image super-resolution using decision tree (SRDT) method is to efficiently classify an input LR patch into one of the leaf nodes and use the corresponding regression model to super-resolve the LR patch into its desired HR patch.

### A. Learning the Super-Resolution Decision Tree

During training, the LR image is first upsampled to the same size as the original HR image using bicubic interpolation. As the bicubic interpolated image has the same size of the HR image, and we call it the  $\mathbf{H}^0$  image. The root node is initialized with all the extracted training LR–HR patch pairs. For each nonleaf node with sufficient training data, we try to find an appropriate binary test that achieves the highest non-negative error reduction among a set of candidate binary tests to split the training data into its left or right child node. However, when the appropriate binary test cannot be found or the number of training data at that node is less than two times the minimum number of training data in a leaf node  $2 \times N_{\min}$ , this node will be declared as a leaf node. A nonleaf node stores the learned binary test and a leaf node stores the learned regression model.

1) *Feature for Regression:* The quality of super-resolved image and the running speed are in relationship to the selected LR feature for regression. The LR features in Zeyde's method [31], ANR method [36] and A+ method [37] are the first- and second-order gradients of the LR patch. To further improve the runtime efficiency, the dimensionality of the feature vector has been reduced by using PCA and preserving 99.9% of the energy. Although multiple features can provide better estimation results, the computational complexity is still much higher than using intensity feature only for estimation. The simple function method [35] uses the normalized LR patch as

input feature. However, patch normalization is not efficient, as the patch mean intensity has to be extracted from the LR patch and then added back to the estimated HR patch. As this paper aims at fast image SR with good quality, we directly use the intensity values of the LR patch as the feature for HR patch estimation to simplify feature extraction process and reduce the complexity of inference. The training data are in the form of LR–HR patch pairs  $\{(\mathbf{x}, \mathbf{y})\}$ , where  $\mathbf{x} \in R^l$  is the vectorized LR patch sampled from the bicubic interpolated image  $\mathbf{H}^0$  and  $\mathbf{y} \in R^h$  is the corresponding vectorized HR patch sampled from the original HR image  $\mathbf{H}$ . Since the bicubic interpolation works very well on the smooth regions, only the patches with edge pixels will be processed. The edge pixels are determined by the Canny edge detector with a threshold of 20.

2) *Learning:* To adapt to varying intensity scales, the binary test adopted in this paper is specified by a set of three parameters,  $\theta = \{P_1, P_2, \text{ and } \tau\}$ . The first two parameters  $P_1$  and  $P_2$  represent two positions on the vectorized LR patch and  $\tau$  is a threshold value. To achieve sufficient randomness, all the parameters of the candidate binary tests are randomly generated. The difference of the intensity values between positions  $P_1$  and  $P_2$  is invariant to intensity changes. This kind of binary test can classify patches that may have different mean intensity but with similar appearance into the same leaf node. The split function  $h(\mathbf{x}, \theta)$  checks the vectorized LR patch  $\mathbf{x}$  according to the binary test  $\theta$  and returns 0 when the requirement fulfills and vice versa

$$h(\mathbf{x}, \theta) = \begin{cases} 0, & \text{if } \mathbf{x}(P_1) < \mathbf{x}(P_2) + \tau \\ 1, & \text{otherwise.} \end{cases} \quad (2)$$

A binary test tries to exclusively divide the training data at the current node into two child nodes according to the result of split function. The training data with return value 0 will be mapped to the left child node and otherwise to the right child node. We evaluate the goodness of a binary test at node  $j$  by the amount of error reduction  $R_j$ , which is the difference between the fitting error at current node  $E_j$  and the weighted fitting error at its two child nodes  $E_L$  and  $E_R$

$$R_j = E_j - \sum_{i \in \{L, R\}} \frac{N_i}{N_j} E_i \quad (3)$$

where  $N_j$  is the number of training data at node  $j$  and  $N_i$ ,  $i = \{L, R\}$ , is the number of training samples at the left or the right child node. The fitting error  $E_j$  at node  $j$  is measured by the mean squared error between the original HR patches  $\mathbf{Y} \in R^{N_j \times h}$  and the reconstructed HR patches  $\mathbf{Y}^R \in R^{N_j \times h}$ , where  $N_j$  is the number of training samples at node  $j$

$$E_j = \frac{1}{N_j} \|\mathbf{Y} - \mathbf{Y}^R\|_2^2. \quad (4)$$

At node  $j$ , a regression model  $\mathbf{C}_j \in R^{l \times h}$  is learned using regularized linear regression to model the relationship between the HR patches  $\mathbf{Y}$  and LR patches  $\mathbf{X} \in R^{N_j \times l}$

$$\mathbf{C}_j = \arg \min_{\mathbf{C}_j} \|\mathbf{Y} - \mathbf{X}\mathbf{C}_j\|_2^2 + \lambda \|\mathbf{C}_j\|_2^2 \quad (5)$$

**Algorithm 1** SRDT Training Algorithm

---

**Input:** HR training images  $\mathbf{H}$  and LR training images  $\mathbf{L}$

1. The initially super-resolved image of  $\mathbf{L}$  is the bi-cubic interpolated image  $\mathbf{H}^0$
2. Extract LR-HR patch pairs with edge pixels from  $\mathbf{H}^0$  and  $\mathbf{H}$  and initiate the root node with all the training data
3. **For** each unprocessed non-leaf node  $j$ 
  4. **If** the training data size  $N_j < 2 \times N_{min}$ 
    5. Declare this node as leaf node and learn the regression model  $\mathbf{C}_j$
  - Else**
    6. Randomly generate  $Q$  binary tests which fulfill (8)
    7. **If** the highest error reduction is positive
      8. Split the training data at node  $j$  into its left and right child node using the learned binary test  $\theta_j$
      9. Declare this node as non-leaf node and store the learned binary test  $\theta_j$
    - Else**
      10. Declare this node as leaf node and learn the regression model  $\mathbf{C}_j$
  - End if**
- End for**

**Output:** Super-Resolution Decision Tree  $DT$

---

where  $\lambda$  is the regularization parameter to enhance the generalization ability of  $\mathbf{C}_j$  and  $\lambda = 0.01$  can generally give a good performance and increase around 0.1 dB in PSNR compared with  $\lambda = 0$ .

Equation (5) has a closed-form solution

$$\mathbf{C}_j = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{Y}. \quad (6)$$

The reconstructed HR image patches  $\mathbf{Y}^R$  are predicted using the learned regression model

$$\mathbf{Y}^R = \mathbf{X} \mathbf{C}_j. \quad (7)$$

In the fast image interpolation via random forests (FIRF) method [50], a constraint on the training data size of two child nodes was proposed to improve the quality of the learned decision trees. In this paper, we adopt this setting. All the binary split should fulfill

$$\max(N_L, N_R) \times k \leq \min(N_L, N_R) \quad (8)$$

where the constraint parameter  $k = 0.7$  follows the setting in FIRF, and  $N_L$  and  $N_R$  are the training data sizes of the left and right child nodes, respectively.

At node  $j$ , the binary test with the highest error reduction from  $Q$  (adopted as the number of elements of the feature vector) randomly generated binary tests  $\Theta = \{\theta_i | i = 1, \dots, Q\}$  is selected as the learned binary test  $\theta_j$ . If no positive error reduction is achieved, that node will not be further divided and will be declared as a leaf node. By recursively partition the training LR–HR patch pairs at nonleaf nodes into leaf nodes, the decision tree can be gradually constructed. The training algorithm of the proposed SRDT method is summarized in Algorithm 1.

**Algorithm 2** SRDT Testing Algorithm

---

**Input:** Low-resolution image  $\mathbf{L}$

1. The initially super-resolved image of  $\mathbf{L}$  is the bi-cubic interpolated image  $\mathbf{H}^0$
2. Extract LR patches with edge pixels from  $\mathbf{H}^0$
3. **For** each LR patch  $\mathbf{x}$ 
  4. **For** each non-leaf node  $j$ 
    5. Partition  $\mathbf{x}$  to left or right child node according to the learned binary test  $\theta_j$ , until reach a leaf node
  - End for**
  6. Predict the HR patch using the regression model  $\mathbf{C}_j$  at the reached leaf node as (9)
- End for**
7. Reconstruct the final super-resolved image  $\mathbf{H}$  by overlapping the predicted HR patches

**Output:** Super-resolved image  $\mathbf{H}$

---

*B. Super-Resolution With Learned Decision Tree*

Algorithm 2 shows the testing algorithm of the proposed SRDT method. Each patch with edge pixels  $\mathbf{x}$  in the bicubic initially upsampled image is recursively partitioned into a left or right child node according to the result of the split function  $h(\mathbf{x}, \theta)$  with the learned binary test until it reaches a leaf node. The reconstructed HR patch  $\mathbf{y}^R$  is obtained by multiplying the LR patch  $\mathbf{x}$  with the reached regression model  $\mathbf{C}$ . Neighboring reconstructed HR patches are overlapped to form the final super-resolved image

$$\mathbf{y}^R = \mathbf{x} \mathbf{C}. \quad (9)$$

*C. Relevant Leaf Nodes in Super-Resolution Decision Tree*

It is good to have all kinds of LR–HR patch pairs with various appearances for training. We find that for a leaf node (leaf<sub>0</sub>) in the learned decision tree, there are three relevant leaf nodes, leaf<sub>*i*</sub> ( $i = 1, 2, 3$ ), where the reached training LR–HR patch pairs are approximately the  $i \times 90^\circ$  rotated versions of them at leaf<sub>0</sub>, and another four relevant leaf nodes leaf<sub>*i*</sub> ( $i = 4, \dots, 7$ ), where the reached training LR–HR patch pairs are approximately the flipped versions (flip down from up) of them at leaf<sub>*i*</sub> ( $i = 0, \dots, 3$ ). Thus, after transformation, the regression models at leaf<sub>*i*</sub> ( $i = 1, \dots, 7$ ) can have very similar coefficients as those at leaf<sub>0</sub>. Based on this observation, we propose two ideas to further improve the efficiency and effectiveness of our proposed SRDT method.

1) *Reserve 1/8 of the Leaf Nodes:* Removing 87.5% of the leaf nodes in a decision tree makes it require only 12.5% of its original size. This could be significant for practical applications. One may argue that the correspondences between leaf nodes in a decision tree are not always one to one. Thus, we usually cannot remove leaf nodes in a simply way. This could be a future research direction.

2) *Fuse Regression Models:* To achieve a better SR quality, combining relevant regression models in a single decision tree is a promising approach as random forests have been proven to be more robust than the decision tree.

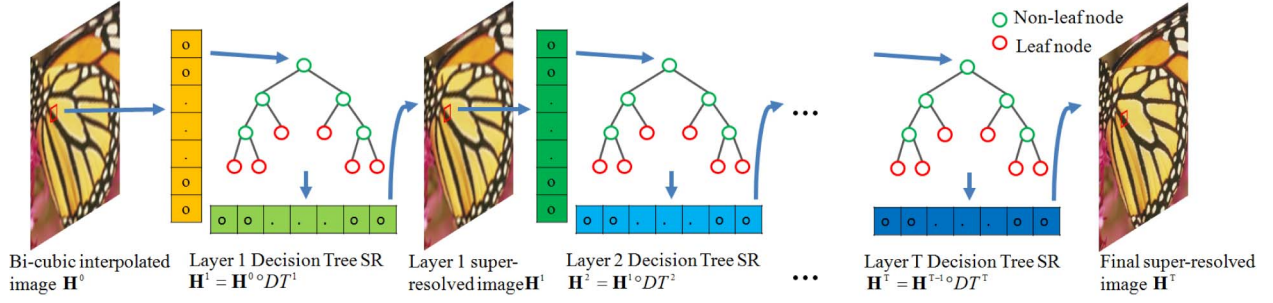


Fig. 1. Flowchart of the proposed image SRHDT.

For an input LR patch vector  $\mathbf{x}_0 \in R^l$  with a dimensionality of  $l$ , we can obtain its  $i \times 90^\circ$  rotated version  $\mathbf{x}_i$  ( $i = 1, 2, 3$ ). The  $m$ th element of  $\mathbf{x}_0$  is correspondingly the  $f_r^i(m)$ th element of  $\mathbf{x}_i$  ( $f_r^i(m)$  means applying the function  $f_r(\cdot)$   $i$  times to  $m$ , e.g.  $f_r^2(m) = f_r(f_r(m))$ )

$$\mathbf{x}_i(f_r^i(m)) = \mathbf{x}_0(m) \quad (10)$$

$$f_r(m) = l - \sqrt{l} \times (m \% \sqrt{l} + 1) + m / \sqrt{l}. \quad (11)$$

The flipped version  $\mathbf{x}_{fi}$  of  $\mathbf{x}_i$  ( $i = 1, 2, 3$ ) can be obtained

$$\mathbf{x}_{fi}(f_f(m)) = \mathbf{x}_i(m) \quad (12)$$

$$f_f(m) = \sqrt{l}[\sqrt{l} - (m / \sqrt{l} + 1)] + m \% \sqrt{l}. \quad (13)$$

$\mathbf{x}_i$  and  $\mathbf{x}_{fi}$  (with  $i = 0, \dots, 3$ ) reach eight different leaf nodes in the learned decision tree. Except  $\mathbf{x}_0$  with the retrieved regression model  $\mathbf{C}_0$ , other retrieved regression models  $\mathbf{C}_i$  (with  $i = 1, 2, 3$ ) and  $\mathbf{C}_{fi}$  (with  $i = 0, \dots, 3$ ) will be manipulated into the form that is suitable to process patch  $\mathbf{x}_0$

$$\mathbf{C}_i^M(f_r^i(m), f_r^i(j)) = \mathbf{C}_i(m, n) \quad (14)$$

$$\mathbf{C}_{fi}^M(f_f(f_r^i(m)), f_f(f_r^i(n))) = \mathbf{C}_{fi}(m, n). \quad (15)$$

For HR patch prediction, the regression model used in Step 6 of Algorithm 2 is the average of some or all manipulated regression models

$$\mathbf{y}^R = \mathbf{x} \times \frac{1}{N} \sum_{k=1}^N \mathbf{C}_k. \quad (16)$$

Regression model fusion with a decision tree idea has three merits: 1) save up to eight times storage space over random forests as a single decision tree is equivalent to eight ones; 2) improve training data usage efficiency because we can put all training data into a decision tree rather than using a subset as in random forests; and 3) improve testing performance over a decision tree.

### III. IMAGE SUPER-RESOLUTION USING HIERARCHICAL DECISION TREES

Although a large number of leaf nodes reduces ambiguity and improves the HR patch prediction accuracy, the training efficiency of SRDT method drops as the training data size increases. Using more training data to obtain a bigger SR decision tree may not be an efficient and effective way to get a better image SR quality. Besides, the size of the learned decision tree is exponentially increasing as the depth of the

decision tree ascends. This could hinder the feasibility of the proposed algorithm.

In SRRF method [43], the random forests approach, which is an ensemble of decision trees, is adopted for image SR. By combining the regression models from multiple decorrelated decision trees, a more robust regression model can be obtained for HR patch prediction. However, the improvement in the SR quality is minor compared with the increment of computational time. With one more decision tree in the random forests, the average computational time increases about 50 ms, while the computational time of using a single decision tree is around 100 ms and the saturated PSNR of the SR random forests with four decision trees is only less than 0.1 dB higher than that of a single decision tree. From both efficiency and effectiveness aspects, we cannot get further improvement (in the direction for the exploitation of fast and high-quality image SR algorithm) from the random forests structure.

We propose in this paper to perform the SRHDT framework to further boost the image SR quality of the proposed SRDT method with short computational time. Fig. 1 presents the flow diagram of the proposed SRHDT method. The general idea is that each layer of the SRHDT framework pushes the estimated HR patch closer to the ground-truth HR patch. The SRHDT method progressively refines the initial bicubic interpolated image using hierarchical decision trees  $HDT = \{DT^1, \dots, DT^T\}$ , where  $T$  is the number of layers in the hierarchical decision trees, and  $DT^i$  is the decision tree in layer  $i$  which is trained using the LR-HR patch pairs from the previous layer super-resolved images and the original HR images of a set of training images. Layer  $i$  decision tree predicts the HR image  $\mathbf{H}^i$  and is denoted by

$$\mathbf{H}^i = \mathbf{H}^{i-1} \circ DT^i. \quad (17)$$

The operation  $\circ$  consists of LR patch extraction, HR patch prediction, and HR image reconstruction corresponding to the SRDT method testing algorithm of Algorithm 2. The learned decision tree in each layer can also follow the regression model fusion strategy as described in Section II-C2, by which the desired HR patch of each LR patch is predicted according to the fused regression model using up to eight relevant regression models in a decision tree for higher accuracy. We denote the SRHDT method with the fused regression model by SRHDT\_f.

The training procedure of the decision tree in each layer in the SRHDT follows the description of Algorithm 1 using the LR patch from the super-resolved images applied by the

**Algorithm 3** SRHDT Training Algorithm

---

**Input:** There are  $T$  sets of HR training images  $\mathbf{H}_t$  and LR training images  $\mathbf{L}_t$ , for  $t = \{1, \dots, T\}$

1. The initially super-resolved image of input  $\mathbf{L}_t$  is the bi-cubic interpolated image  $\mathbf{H}_t^0$
2. Train Layer 1 Super-Resolution Decision Tree  $DT^1$  using  $\mathbf{L}_1$  and  $\mathbf{H}_1$
3. **For**  $t = 2$  to  $T$ 
  4. **For**  $i = 1$  to  $t - 1$ 
    5.  $\mathbf{H}_t^i = \mathbf{H}_t^{i-1} \circ DT^i$
- End for**
6. Train the Layer  $t$  Super-Resolution Decision Tree  $DT^t$  using  $\mathbf{H}_t^{t-1}$  and  $\mathbf{H}_t$
- End for**

**Output:** Hierarchical Decision Trees  $HDT = \{DT^1, \dots, DT^T\}$

---

**Algorithm 4** SRHDT Testing Algorithm

---

**Input:** Low-resolution image  $\mathbf{L}$

1. The initially super-resolved image of  $\mathbf{L}$  is the bi-cubic interpolated image  $\mathbf{H}^0$
2. **For**  $t = 1$  to  $T$ 
  3. Super-resolve image  $\mathbf{H}^t = \mathbf{H}^{t-1} \circ DT^t$
- End for**

**Output:** Super-resolved image  $\mathbf{H}^T$

---

previous layer decision trees and the HR patches from the original HR training images. The training algorithm of the SRHDT method is shown in Algorithm 3.

With the learned hierarchical decision trees  $HDT = \{DT^1, \dots, DT^T\}$ , the initially bicubic interpolated input LR image  $\mathbf{H}^0$  will be enhanced layer by layer. Algorithm 4 shows the testing algorithm of the proposed SRHDT method.

The proposed SRHDT is similar to the spirit in the deep-learning-based image SR method, SRCNN, which has linear transformation and nonlinear mapping. In SRHDT, the HR image patch prediction performed on a leaf node is a linear transformation and the patch overlapping is a nonlinear mapping that clips the dynamic range of pixel intensity between 0 and 255. Multiple layers in SRHDT could correspond to multiple linear transformations and nonlinear mapping structures in the deep learning model.

#### IV. DATA-DEPENDENT MODEL FOR VIDEO SUPER-RESOLUTION

For single-image SR, it is difficult to learn a complete model, which can provide superb SR quality for variant image contents. Without the prior knowledge of the image content, the ambiguity between LR patches in training images impedes us to get better SR results. Video SR is in a different scenario as single-image SR. There are multiple similar images of the same scene in a video. If a few HR images (for example, I frames) are provided for each scene, a data-dependent model can learn from the provided LR–HR image pairs. With the

learned data-dependent model, the desired HR images of the input LR images (for example, B frames) within the same scene can be predicted more effectively.

Applying image SR technique for video compression and video coding is a promising research direction. However, these applications require the minimization of the difference between the original HR images and the super-resolved images. The existing general image SR models are not good enough to accomplish this task. Designing a learning-based SR method with a data-dependent model could be one of the possible solutions.

All the learning-based SR methods including the proposed SRDT method and SRHDT method can be converted into a dependent model using the provided LR–HR image pairs. Different from the general model, which can be learned offline without considering the training time, the training speed for the data-dependent model should be as fast as possible. In SRDT and SRHDT methods, much training time is used to construct the regression models for error reduction evaluation. In the data-dependent model for video SR, we propose to randomly select binary tests without error reduction evaluation, which could slightly sacrifice the SR quality. However, the image SR quality using data-dependent SRDT and SRHDT is much better than using the learned general SRDT and SRHDT. The only difference between the data-dependent SRDT training algorithm and Algorithm 1 is that in the data-dependent algorithm if there are enough training data for further split in a node, a randomly generated binary test fulfilling (8) will be selected as the learned binary test without error reduction evaluation.

The data-dependent model SRDT training algorithm is around 80 times faster than the original general model SRDT training algorithm. The average PSNR has only around a 0.1-dB decrease.

#### V. EXPERIMENTAL RESULTS

In this section, we report the testing results and analyze the proposed the SRDT method, the SRHDT method, as well as the SRHDT\_f method with the upscaling factors of 2 and 3. We adopt the PSNR and structural similarity index (SSIM) [51] to evaluate the image SR quality for all methods. For color image, we converted the image from RGB color space to YCbCr color space (Y-axis represents luminance, and Cb axis and Cr axis are chrominance channels) and performed SR only on the luminance channel. The chrominance channels were upsampled to the desired size by the bicubic interpolation method. For an upscaling factor of  $p$ , the LR image was obtained by a bicubic filter followed by downsampling by a factor of  $p$  (realized by MATLAB function *imresize*); the HR patch size was selected as  $(3p) \times (3p)$  so that the corresponding LR patch size is always  $3 \times 3$ . All the experimental results were obtained on an Intel Core i7 3.5-GHz processor. The proposed methods were implemented in C++, taking the bicubic interpolated image as input.

##### A. Single-Image Super-Resolution With the General Model

For the general model, we used 118 training images for training (note that the general model refers to the regression



TABLE I

PSNR (dB), SSIM, AND AVERAGE RUNNING TIME (s) RESULTS ON *Set14* BY BICUBIC INTERPOLATION AND THE PROPOSED SRHDT METHOD WITH DIFFERENT NUMBER OF LAYERS FOR UPSCALING FACTORS  $\times 2$

Images	Methods													
	Bi-cubic		SRHDT 1 Layer			SRHDT 2 Layers			SRHDT 3 Layers			SRHDT 4 Layers		
	PSNR	SSIM	PSNR	SSIM	Time	PSNR	SSIM	Time	PSNR	SSIM	Time	PSNR	SSIM	Time
Baboon	24.87	0.6980	25.63	0.7597	0.13	25.73	<b>0.7710</b>	0.24	25.77	0.7698	0.33	<b>25.78</b>	0.7701	0.44
Barbara	27.94	0.8398	<b>28.51</b>	0.8685	0.16	28.35	<b>0.8705</b>	0.30	28.29	0.8694	0.43	28.21	0.8689	0.57
Bridge	26.60	0.7936	27.70	0.8446	0.15	27.83	<b>0.8522</b>	0.27	27.88	0.8516	0.40	<b>27.89</b>	0.8517	0.49
Coastguard	29.21	0.7879	30.50	0.8393	0.05	30.64	<b>0.8480</b>	0.09	30.64	0.8460	0.13	<b>30.66</b>	0.8460	0.17
Comic	25.93	0.8470	28.01	0.9053	0.05	28.44	0.9157	0.09	28.60	0.9175	0.13	<b>28.70</b>	<b>0.9191</b>	0.17
Face	34.78	0.8622	35.57	0.8812	0.03	35.62	<b>0.8837</b>	0.06	<b>35.63</b>	0.8831	0.08	<b>35.63</b>	0.8829	0.11
Flowers	30.28	0.8979	32.63	0.9305	0.08	33.08	0.9355	0.16	33.26	0.9365	0.21	<b>33.37</b>	<b>0.9372</b>	0.28
Foreman	34.67	0.9535	37.60	0.9684	0.03	38.04	0.9702	0.07	<b>38.19</b>	<b>0.9706</b>	0.09	38.18	<b>0.9706</b>	0.12
Lenna	34.64	0.9109	36.40	0.9254	0.09	36.56	0.9270	0.17	36.62	<b>0.9271</b>	0.23	<b>36.64</b>	<b>0.9271</b>	0.31
Man	29.21	0.8454	30.65	0.8804	0.13	30.92	0.8865	0.24	31.00	0.8868	0.33	<b>31.04</b>	<b>0.8872</b>	0.43
Monarch	32.86	0.9598	36.80	0.9746	0.12	37.53	0.9763	0.23	37.77	0.9767	0.32	<b>37.90</b>	<b>0.9770</b>	0.42
Pepper	35.01	0.9071	36.83	0.9174	0.09	37.02	0.9185	0.16	37.09	0.9187	0.23	<b>37.12</b>	<b>0.9188</b>	0.31
Ppt3	26.79	0.9439	30.08	0.9764	0.09	30.79	0.9807	0.18	31.08	0.9822	0.26	<b>31.27</b>	<b>0.9830</b>	0.34
Zebra	30.57	0.9090	33.63	0.9412	0.12	33.68	<b>0.9429</b>	0.24	33.78	0.9427	0.32	<b>33.80</b>	0.9426	0.42
Average	30.24	0.8683	32.18	0.9009	0.09	32.45	0.9056	0.18	32.54	0.9056	0.26	<b>32.58</b>	<b>0.9059</b>	0.33



Fig. 2. Parts of the training images.

models that can be used to construct an SR of any LR image, i.e., different sets of images are used for training and SR evaluation). All the training images were downloaded from featured pictures in Wikipedia within the title of Plants, and parts of the training images are shown in Fig. 2. The training images can provide around 10 000 000 LR–HR patch pairs for each layer decision tree training. Besides, we also applied the standard 91 training images [27] for training (the results are reported in Table II and marked by \*). *Set5* [27] (with five testing images: *Baby*, *Bird*, *Butterfly*, *Head*, and *Woman*) and *Set14* [31] (with 14 testing images: *Baboon*, *Barbara*, *Bridge*, *Coastguard*, *Comic*, *Face*, *Flowers*, *Foreman*, *Lenna*, *Man*, *Monarch*, *Pepper*, *Ppt3*, and *Zebra*) were used to evaluate for the image SR quality.

1) *Experimental Settings*: The performance of the SRDT is controlled by the quality of the regression models and the number of leaf nodes. The quality of the regression models can be improved by using more training LR–HR patch pairs to construct each of the regression models. The reason behind is obvious. The relationship between LR and HR patches can be more precisely represented using denser sampled training data. Fig. 3(a) shows the relationship between the minimum number of training data in a leaf node  $N_{\min}$  and the average PSNR of the super-resolved images in *Set14* when the leaf node number is around 256 and the upscaling factor is 2. As  $N_{\min}$  increases from 1 to 128 times of the minimum number of LR–HR patch pairs required to estimate the regression model  $N_{\text{est}}$  (where the patch size used is  $6 \times 6$  and the minimum number is 36), the average PSNR of the test images has around 1.4 dB improvement. In the following experiments,

we adopted  $N_{\min}$  as 50 times of  $N_{\text{est}}$ . The number of leaf nodes is a key factor that affects the image SR accuracy. The basic idea of recent fast image SR methods [35]–[44] is to divide the image patch space into numerous subspaces and use simple linear model to relate LR and HR patches. The number of leaf nodes thus affects the subspace size. If the subspace is too big, the assumption that a simple linear model can represent the LR–HR relationship in each subspace would no longer hold. Fig. 3(b) shows the PSNR of the test images in *Set14* with respect to different number of leaf nodes. We vary the number of leaf nodes from 1 to around 2048 using the number of training data varying from 2500 to 5 120 000. It can be observed from Fig. 3(b) that there is an improvement in PSNR of around 0.1 dB when the number of leaf nodes (the size of training data) doubles. From the SRDT perspective, although learning large enough decision tree is beneficial, an exponential increase in training time and training data would become unaffordable.

To accommodate the huge training time and training data for a big SRDT, we proposed the SRHDT method. We rotated the HR training images with certain angles to generate enough training data to form multiple layers in the SRHDT method. Table I presents the PSNR, SSIM, and running time of each layer in SRHDT (totally four layers) on *Set14* [31] for an upscaling factor of 2. The decision tree in each layer had around 3800 leaf nodes. The running time of the SRHDT method increases linearly as the increment of layer number in the SRHDT. The SRHDT with four layers achieves the highest PSNR and SSIM on most of the testing images in *Set14*. We can find that there is an improvement of 0.26, 0.10, and 0.04 dB in PSNR for layers 2, 3, and 4, respectively. The overall PSNR improvement compared with the layer 1 result is 0.40 dB. With the same PSNR improvement, the SRDT may need around 16 times more training data (under the assumption that when training data double, there is around 0.1-dB improvement).

2) *Comparison With State-of-the-Art Algorithms*: As stated in Section II-C2, our proposed hierarchical framework with the fused regression model (SRHDT\_f) method can achieve even

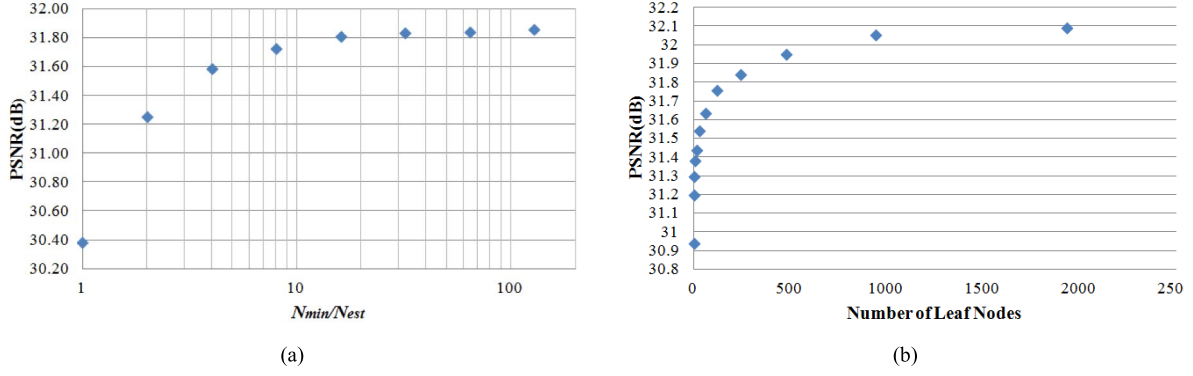


Fig. 3. Upscaling factor = 2. (a) Relationship between  $N_{\min}$  and the average PSNR of the super-resolved images in Set14 when magnification factor is 2, with the number of leaf nodes = 256. (b) Relationship between the number of leaf nodes and the average PSNR of the super-resolved images in Set14 with  $N_{\min} = 1800$ .

TABLE II

PSNR (dB), SSIM, AND AVERAGE RUNNING TIME (s) BY DIFFERENT IMAGE SR METHODS FOR UPSCALING FACTORS  $\times 2$  AND  $\times 3$  ON Set5 and Set14 (THE BEST RESULT IN EACH ROW IS MARKED IN RED AND THE SECOND BEST RESULT IS MARKED IN BLUE)

Dataset	Scale	Eval. Mat	Bi-cubic	Zeyde's [31]	A+ [37]	SRCNN [41]	Peleg's [33]	SRRF [43]	Proposed SRHDT *	Proposed SRHDT_f *	Proposed SRDT	Proposed SRHDT	Proposed SRHDT_f
Set5	$\times 2$	PSNR	33.66	35.79	36.56	36.35	36.11	36.17	36.72	36.90	36.32	36.77	36.92
		SSIM	0.9299	0.9494	0.9545	0.9522	0.9504	0.9490	0.9533	0.9543	0.9514	0.9539	0.9546
		Time	0.000	2.570	0.810	4.597	9.770	0.331	0.166	0.440	0.037	0.150	0.410
	$\times 3$	PSNR	30.41	31.94	32.66	32.42	32.36	-	32.93	33.05	32.36	32.94	33.08
		SSIM	0.8686	0.8973	0.9095	0.9024	0.9040	-	0.9109	0.9130	0.9030	0.9114	0.9132
		Time	0.000	1.880	0.670	4.619	18.590	-	0.268	0.570	0.063	0.256	0.555
Set14	$\times 2$	PSNR	30.24	31.84	32.32	32.23	31.99	32.08	32.52	32.63	32.18	32.58	32.67
		SSIM	0.8683	0.8985	0.9054	0.9036	0.9005	0.8969	0.9047	0.9062	0.9009	0.9059	0.9069
		Time	0.000	5.490	1.730	10.382	20.270	0.734	0.373	0.942	0.084	0.339	0.903
	$\times 3$	PSNR	27.54	28.67	29.16	29.03	28.95	-	29.34	29.42	28.97	29.38	29.46
		SSIM	0.7726	0.8067	0.8182	0.8128	0.8129	-	0.8193	0.8214	0.8127	0.8203	0.8220
		Time	0.000	2.960	1.050	10.377	44.060	-	0.587	1.344	0.137	0.553	1.230

better SR results by fusing regression models from relevant leaf nodes within the same decision tree. Table II reports the average PSNR, SSIM, and average running time of the proposed SRDT method, SRHDT method, SRHDT\_f method, SRRF method [43], and the state-of-the-art image SR algorithms including Zeyde's method [31], the A+ method [37], the SRCNN method [41], and Peleg's method [33] for upscaling factors of 2 and 3 on Set5 and Set14. Among the comparison methods, Zeyde's method [31] is the state-of-the-art fast image SR algorithm that is based on OMP for SC and uses PCA for input LR feature dimensionality reduction. The A+ method [37] achieved the best performance in the literature on both SR quality and speed using SC for patch classification and prelearned regression models. The SRCNN method [41] explores applying conventional neural network for SR and realized with high quality and fast speed. Peleg's method [33] applies an efficient feedforward neural network implementation for HR patch sparse representation prediction that leads to fast realization. We have used the realization codes from the authors' websites for comparison to avoid experiment biasing.

Table III shows the parameter settings of our proposed methods: the patch size, the regularization parameter, the constraint parameter, the minimum number of training data in a leaf node, the number of randomly generated binary tests

in a node, the number of layers in hierarchical decision trees, the overlapping pixels, and the number of fused regression models in the SRHDT\_f method. The number of layers was selected as 4, because the improvement is less than 0.05 dB with one additional layer. Partial patch overlapping is adopted for fast testing. Each pixel will have nine estimation values with a patch overlapping of 4 pixels and 6 pixels for upscaling factors of 2 and 3, respectively. Only four regression models are applied for model fusion, because the performance would saturate with around four regression models combined (similar to the result reported in [43]). There are around 3900 and 1800 leaf nodes in each decision tree trained using 118 training images for the upscaling factors of 2 and 3, respectively. The maximum depths of the decision tree are 13 and 12, and the average number of nonleaf nodes checked is 11.91 and 10.84, for the upscaling factors of 2 and 3, respectively. As the training data size is smaller in the standard 91 training dataset [27],  $N_{\min}$  was set to be smaller than that listed in Table III ( $N_{\min}$  is 32 times of  $N_{\text{est}}$ , equal to 1152 and 2592 for the upscaling factors of 2 and 3, respectively) to result in sufficient leaf nodes in the decision tree (around 4900 and 2100 for the upscaling factors of 2 and 3, respectively).

To summarize the quality performance in Table II, let us average the results on Set5 and Set14. For the upscaling factor of 2, the average PSNRs of the proposed SRHDT\_f



TABLE III  
PARAMETER SETTINGS UNDER UPSCALING FACTORS OF 2 AND 3

Scenario	Patch Size	$\lambda$	$k$	$N_{min}$	$Q$	$T$	Overlapping	Number of fuse models
$\times 2$	$6 \times 6$	0.01	0.7	1800	36	4	4 pixels	4
$\times 3$	$9 \times 9$	0.01	0.7	4020	81	4	6 pixels	4

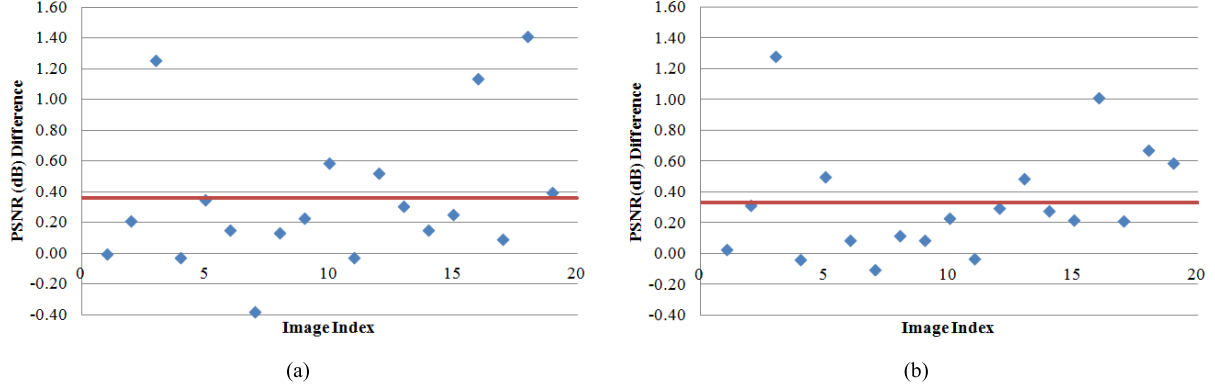


Fig. 4. PSNR (dB) differences between the proposed SRHDT\_f method and the A+ method of all testing images in *Set5* and *Set14* (*Set5* images followed by *Set14* images) for (a) upscaling factor of 2, with an average difference in PSNR of 0.36 dB and (b) upscaling factor of 3, with an average difference in PSNR of 0.33 dB.

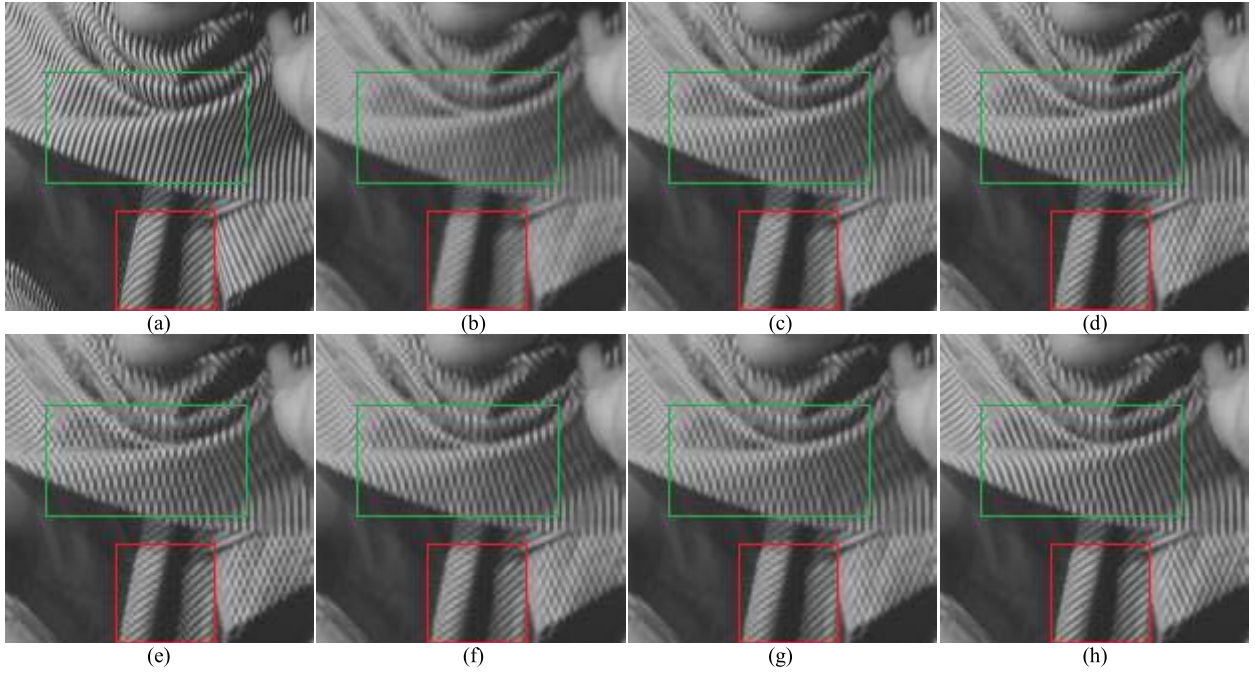


Fig. 5. Reconstructed HR images of *Barbara* from the *Set14* dataset by different SR methods for the upscaling factor of 2. (a) Original HR image. (b) Bicubic (PSNR = 27.94 dB and SSIM = 0.8398). (c) Zeyde's (PSNR = **28.63** dB and SSIM = 0.8717). (d) A+ (PSNR = **28.63** dB and SSIM = 0.8721). (e) SRCNN (PSNR = 28.53 dB and SSIM = **0.8743**). (f) Peleg's (PSNR = 28.48 dB and SSIM = 0.8688). (g) Proposed SRDT (PSNR = 28.51 dB and SSIM = 0.8685). (h) Proposed SRHDT\_f (PSNR = 28.25 dB and SSIM = 0.8701).

method are 0.10, 0.36, 0.47, 0.52, 0.72, 0.75, and 0.91 dB higher than those of the proposed SRHDT method, the A+ method, the SRCNN method, the proposed SRDT method, Peleg's method, the SRRF method, and Zeyde's method, respectively. For the upscaling factor of 3, the average PSNRs of the proposed SRHDT\_f method are 0.10, 0.33, 0.49, 0.55, 0.56, and 0.88 dB higher than those of the proposed SRHDT method, the A+ method, SRCNN method, the proposed SRDT method, Peleg's method, and Zeyde's method,

respectively. The proposed regression model fusion method SRHDT\_f leads the proposed SRHDT method with around 0.1-dB gain without learning extra information. Compared with the state-of-the-art image SR methods, SRHDT\_f has an around 0.3 and 0.5 dB improvement over the A+ method and SRCNN method for both upscaling factors of 2 and 3, respectively. Note that the proposed SRDT method has results comparable with those of the SRCNN method and Peleg's method, while has around 0.3-dB advantage over

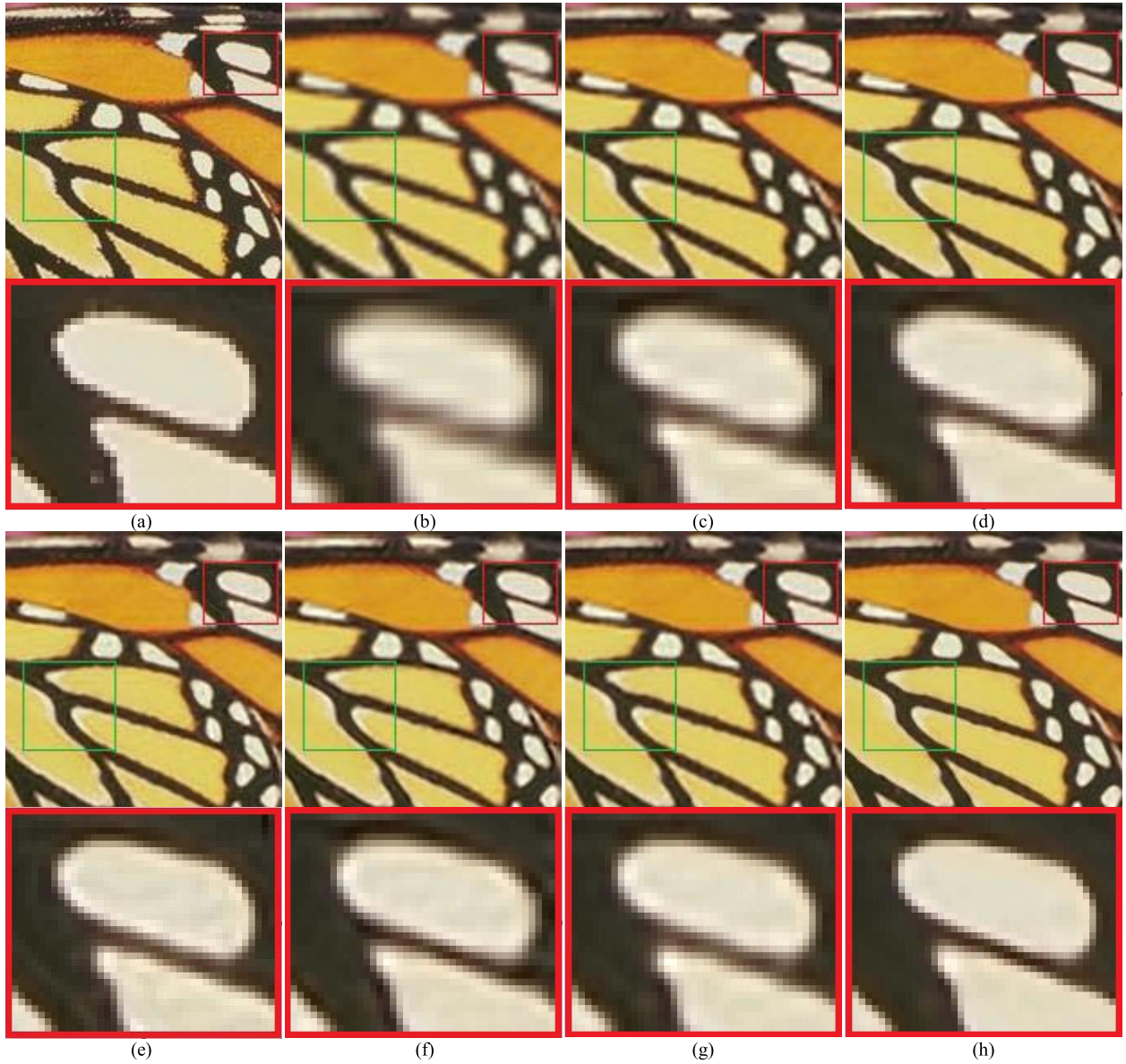


Fig. 6. Reconstructed HR images of *Butterfly* from the *Set5* dataset by different SR methods for the upscaling factor of 3. (a) Original HR image. (b) Bicubic (PSNR = 24.09 dB and SSIM = 0.8241). (c) Zeyde's (PSNR = 26.06 dB and SSIM = 0.8803). (d) A+ (PSNR = 27.46 dB and SSIM = 0.9124). (e) SRCNN (PSNR = 27.76 dB and SSIM = 0.9031). (f) Peleg's (PSNR = 26.92 dB and SSIM = 0.9022). (g) Proposed SRDT (PSNR = 27.07 dB and SSIM = 0.8995). (h) Proposed SRHDT\_f (PSNR = **28.74** dB and SSIM = **0.9280**).

Zeyde's method. To further demonstrate the performance of the proposed SRHDT\_f method, Fig. 4 shows the PSNR (dB) differences between the results of the SRHDT\_f method and the A+ method on all testing images for upscaling factors of 2 and 3, respectively. The proposed SRHDT\_f can provide more than 1-dB gain in PSNR compared with the A+ method on testing images *Butterfly*, *Monarch*, and *Ppt3*. Although the proposed SRHDT\_f method reports lower PSNR on the testing image *Barbara*, which could be due to the strongly aliased patterns on *Barbara* as shown in the green rectangles in Fig. 5. However, our proposed SRHDT\_f method generates a better visual quality with sharper edges, as shown in Fig. 5.

In Figs. 5–8, the subjective performances of the proposed SRDT method, SRHDT\_f method, and other methods

are demonstrated. Significant differences among different methods are highlighted by red and green rectangles. From visual comparisons, we can see that the reconstructed HR images of our proposed SRHDT\_f method are with sharper edges and less artifacts compared with those of other state-of-the-art image SR algorithms.

The runtime efficiency of the proposed SRDT method, SRHDT method, and SRHDT\_f method is a great advantage over that of other methods. The proposed SRDT method is the fastest image SR algorithm in Table II except the bicubic interpolation method. With a comparable SR quality, the proposed SRDT method requires only around 1% and 0.4% processing times of the SRCNN method and Peleg's method. The running time of the proposed SRHDT method is about 4 times longer than that of the proposed SRDT method,



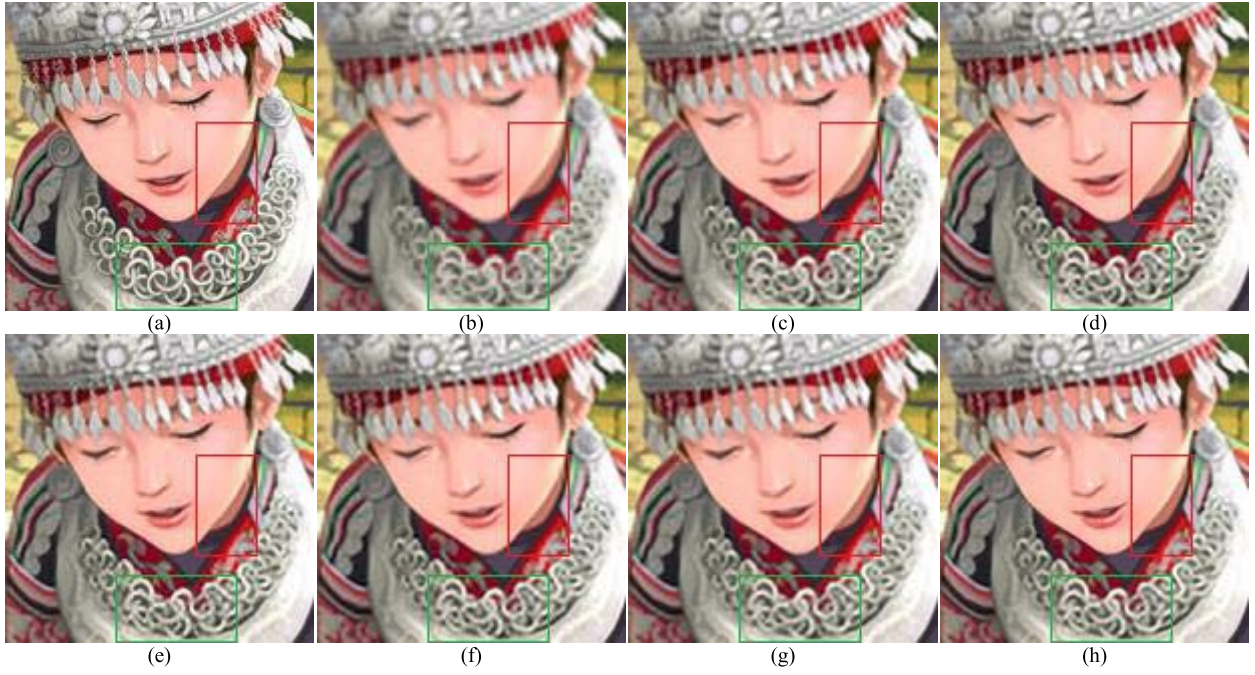


Fig. 7. Reconstructed HR images of *Comic* from the *Set14* dataset by different SR methods for the upscaling factor of 2. (a) Original HR image. (b) Bicubic (PSNR = 25.93 dB and SSIM = 0.8470). (c) Zeyde's (PSNR = 27.55 dB and SSIM = 0.8968). (d) A+ (PSNR = 28.19 dB and SSIM = 0.9118). (e) SRCNN (PSNR = 28.17 dB and SSIM = 0.9099). (f) Peleg's (PSNR = 27.94 dB and SSIM = 0.9045). (g) Proposed SRDT (PSNR = 28.01 dB and SSIM = 0.9053). (h) Proposed SRHDT\_f (PSNR = **28.78** dB and SSIM = **0.9205**).

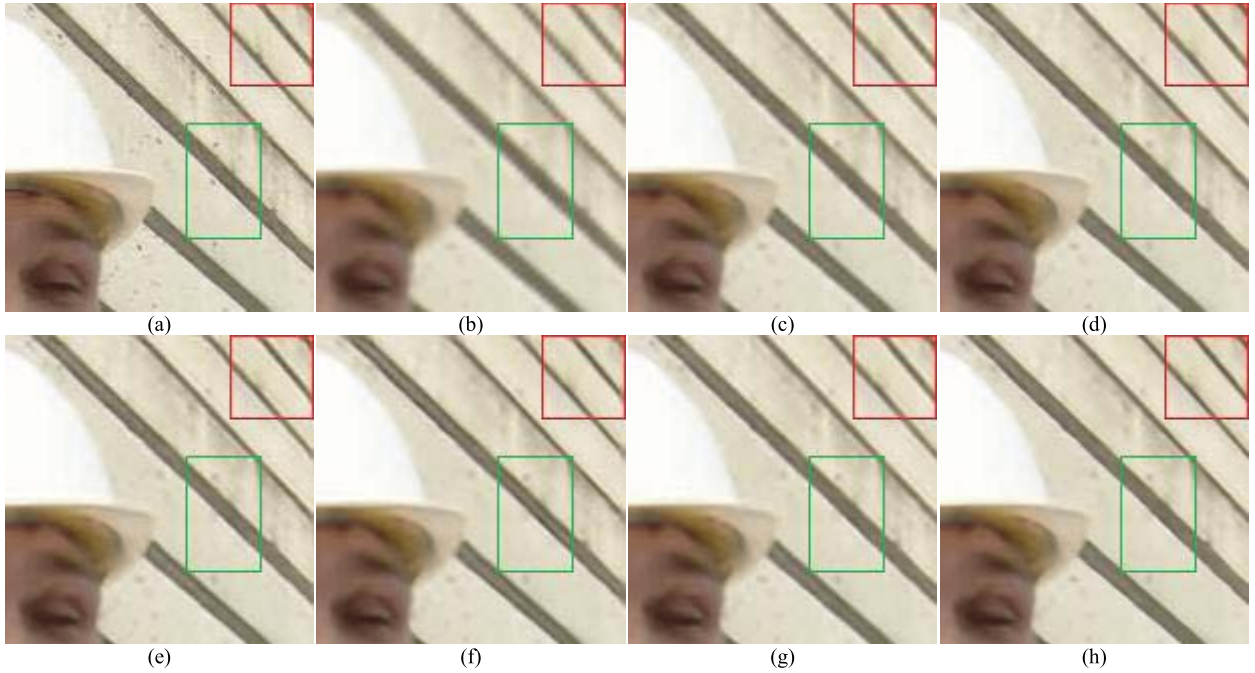


Fig. 8. Reconstructed HR images of *Foreman* from the *Set14* dataset by different SR methods for the upscaling factor of 3. (a) Original HR image. (b) Bicubic (PSNR = 31.96 dB and SSIM = 0.9099). (c) Zeyde's (PSNR = 34.16 dB and SSIM = 0.9323). (d) A+ (PSNR = 35.61 dB and SSIM = 0.9428). (e) SRCNN (PSNR = 34.82 dB and SSIM = 0.9339). (f) Peleg's (PSNR = 35.09 dB and SSIM = 0.9388). (g) Proposed SRDT (PSNR = 34.78 dB and SSIM = 0.9353). (h) Proposed SRHDT\_f (PSNR = **36.09** dB and SSIM = **0.9441**).

since there are four layers in the SRHDT framework. The SRHDT method achieves the second best PSNR in every experimental setting. Moreover, its running speed is two to five times faster than the A+ method, which obtains the third best PSNR results. Fusing regression models from relevant

leaf nodes in a decision tree would make the testing time of the proposed SRHDT\_f method twice as that of the proposed SRHDT method.

Similar to [41], the above comparison on runtime speed made use of MATLAB or C++ platform, which can hardly be

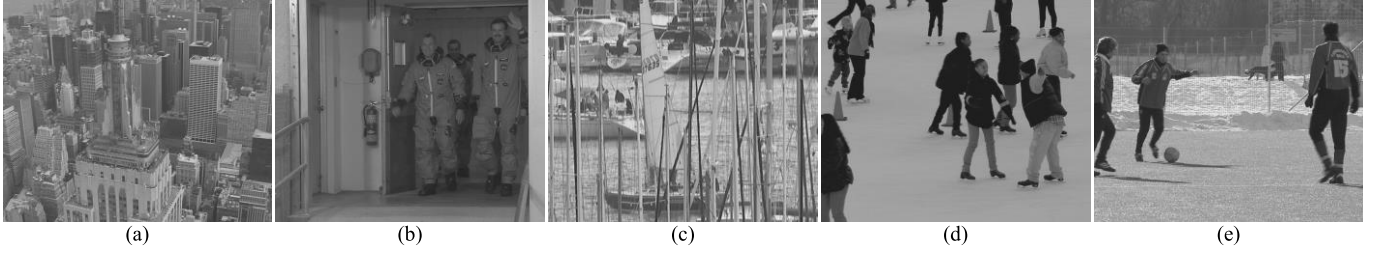


Fig. 9. First image of the testing sequence. (a) City. (b) Crew. (c) Harbour. (d) Ice. (e) Soccer.

TABLE IV

PSNR (dB) AND SSIM RESULTS BY BICUBIC INTERPOLATION, THE PROPOSED GENERAL MODEL SRDT METHOD, THE PROPOSED SRHDT\_f METHOD, AND THE PROPOSED DATA-DEPENDENT MODEL SRDT METHOD [WITH TRAINING TIME (MIN)] ON FIVE TESTING SEQUENCES FOR A UPSCALING FACTOR OF 2

Sequences	Methods								
	Bi-cubic		General SRDT Method		General SRHDT f Method		Data Dependent SRDT Method		
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	Training Time
<i>City</i>	29.11	0.8558	31.18	0.9060	32.13	0.9170	<b>33.08</b>	<b>0.9179</b>	5.52
<i>Crew</i>	36.51	0.9392	38.77	0.9524	38.85	0.9540	<b>39.59</b>	<b>0.9547</b>	2.36
<i>Harbour</i>	31.83	0.9314	35.10	0.9631	34.76	0.9641	<b>35.87</b>	<b>0.9661</b>	6.11
<i>Ice</i>	35.58	0.9578	38.82	0.9682	39.41	<b>0.9697</b>	<b>39.43</b>	0.9693	0.78
<i>Soccer</i>	31.24	0.8707	33.05	0.9113	<b>33.25</b>	0.9164	33.11	<b>0.9116</b>	4.66

completely fair. Let us analyze the computational complexity of the A+ method [37], SRCNN method [41], and our proposed SRDT method and SRHDT method. Let  $M$  be the number of pixels in the bicubic interpolated image,  $\alpha$  be the ratio of processed patches by patch overlapping,  $p$  be the dimensionality of the extracted LR features,  $K$  be the number of mapping functions, and  $n$  be the dimensionality of the HR patches. The computational complexity of A+ method is

$$O(\alpha M(pm + mK + mn + 1))$$

where  $m$  is the reduced dimensionality of the LR features (for an upscaling factor of 3,  $\alpha = 1/9$ ,  $p = 324$ ,  $m = 30$ ,  $K = 1024$ , and  $n = 81$ ).

The computational complexity of the SRCNN method is

$$O(M(n_1(f_1^2 + 2) + n_2(n_1 f_2^2 + 2) + n_2(f_3^2 + 2)))$$

where  $f_i$  is the spatial size of the filter with  $i = 1, 2, 3$ , and  $n_j$  is the number of filters in the  $j$ th level (for the upscaling factor of 3,  $f_1 = 9$ ,  $f_2 = 1$ ,  $f_3 = 5$ ,  $n_1 = 64$ , and  $n_2 = 32$ ).

The computational complexity of our SRDT method is

$$O(\alpha\beta M(\log_2 K + pn + 1))$$

and the computational complexity of our SRHDT method is

$$O(\alpha\beta TM(\log_2 K + pn + 1))$$

where  $\beta$  is the ratio of edge pixels processed by our method,  $T$  is the number of layers in the hierarchical decision trees (for the upscaling factor of 3,  $\alpha = 1/9$ ,  $\beta \approx 0.7$ ,  $K = 1800$ ,  $p = 81$ ,  $n = 81$ , and  $T = 4$ ).

From the above analysis, the computational complexity of SRCNN is about 16.2 times and 4.1 times as that of the proposed SRDT and SRHDT methods, respectively. The computational complexity of the A+ is about 9.3 times

and 2.3 times as that of the proposed SRDT and SRHDT methods, respectively.

#### B. Video Super-Resolution With the Data-Dependent Model

In this section, we evaluate the proposed SRDT for video SR with the data-dependent model using five video sequences *City*, *Crew*, *Harbour*, *Ice*, and *Soccer* for upscaling factor of 2 (the first image of each sequence is shown in Fig. 9). All these sequences are with spatial resolution of  $704 \times 576$  and a frame rate of 30 frames/s. For each video sequence, the first frame and the 31th frame were selected as training data, and the second frame to the 30th frame were testing frames. Since the training data are not sufficient, it is the first priority to obtain more leaf nodes in the learned decision tree. The minimum number of training data in a leaf node  $N_{\min}$  was set to  $2 \times N_{\text{est}}$  and the regularization parameter  $\lambda$  was set to 1. The SRDT method has to be evaluated not only by the quality assessment metrics (PSNR and SSIM) but also by the training time, because a too long training time is unacceptable for a video application that requires a huge number of frames.

Table IV shows the average PSNR and SSIM of the bicubic interpolation, the proposed general model SRDT method, the proposed general model SRHDT\_f method, and the proposed data-dependent SRDT method (with the training time in minutes) on five testing video sequences. We can find that the proposed data-dependent SRDT method achieves around 1.0-dB and 1.9-dB gain in PSNR compared with the proposed general model SRHDT\_f method and the proposed general model SRDT method on the testing sequence *City*. Fig. 10 shows that the aliasing region in the 16th frame of the testing sequence *City* has been better recovered by the data-dependent model SRDT compared with the general model methods. As discussed in Section V-A2, the running time of the SRDT method is around nine times faster than the

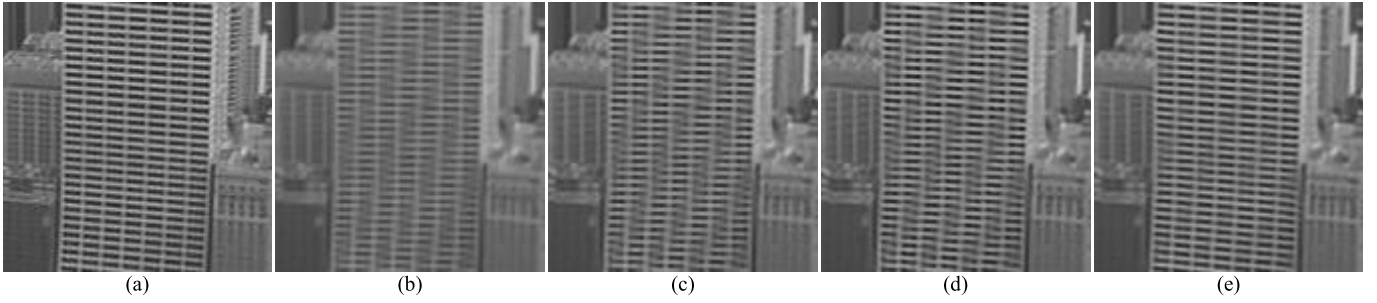


Fig. 10. Reconstructed HR images of the 16th frame in the testing sequence *City* by different SR methods for the upscaling factor of 2. (a) Original HR image. (b) Bicubic (PSNR = 28.18 dB and SSIM = 0.8336). (c) General model SRDT (PSNR = 30.11 dB and SSIM = 0.8885). (d) General model SRHDT\_f (PSNR = 31.02 dB and SSIM = 0.9011). (e) Data-dependent SRDT (PSNR = 31.89 dB and SSIM = 0.9012).

SRHDT\_f method. Thus, the data-dependent model, SRDT method, can greatly reduce the prediction error while with much lower running complexity. In the experimental results, the sequence *Soccer* is an exceptional case in which the data-dependent model obtains lower PSNR compared with the general model. The main reason could be that there were too many movements of the players in the sequence to be captured by the first frame and 31st frame, such that the learned data-dependent model cannot effectively super-resolve other frames.

## VI. CONCLUSION

In this paper, we have proposed a novel approach for efficient and effective single-image super-resolution using decision tree and hierarchical decision trees. We suggested that a combination of the classification process and regression process should be an efficient solution. The proposed SRDT method enables us to exploit fast image patch classification by comparing the intensity values of several pixel pairs, which has very low complexity. The linear regression model using simple feature keeps the computation cost of mapping LR patch to its desired HR patch low. To further boost the performance and improve the training efficiency of using single decision tree for image SR, a hierarchical decision trees framework (SRHDT method) has been proposed. The decision tree in each layer of the hierarchical framework consistently refines the SR results of the previous layer. The overall improvement of the hierarchical decision trees framework is over 0.4 dB in PSNR compared with the result of our proposed SRDT method. As each training LR–HR patch pair can find its approximately rotated and flipped versions, we have proposed to fuse the regression models from up to eight relevant leaf nodes in a decision tree to form a more robust regression model for HR patch prediction that provides another around 0.1 dB PSNR gain. Our extensive experimental results show that our proposed SRDT method achieves SR quality comparable with those of the sparse-representation-based method (Peleg’s method) and the deep-learning-based method (the SRCNN method), while ours is much faster. Furthermore, our proposed SRHDT\_f method generates more than a 0.3-dB higher PSNR compared with the result of the state-of-the-art fast image SR method A+.

For future work, some preliminary extension work of the data-dependent model has shown a promising result, which could help to further improve the current video SR methods or lead a path for video compression/coding.

## REFERENCES

- [1] R. G. Keys, “Cubic convolution interpolation for digital image processing,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 29, no. 6, pp. 1153–1160, Dec. 1981.
- [2] H. S. Hou and H. C. Andrews, “Cubic splines for image interpolation and digital filtering,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 26, no. 6, pp. 508–517, Dec. 1978.
- [3] T. Blu, P. Thévenaz, and M. Unser, “Linear interpolation revitalized,” *IEEE Trans. Image Process.*, vol. 13, no. 6, pp. 710–719, May 2004.
- [4] T. M. Lehmann, C. Gönnner, and K. Spitzer, “Addendum: B-spline interpolation in medical image processing,” *IEEE Trans. Med. Imag.*, vol. 20, no. 7, pp. 660–665, Jul. 2001.
- [5] X. Li and M. T. Orchard, “New edge-directed interpolation,” *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1521–1527, Oct. 2001.
- [6] W.-S. Tam, C.-W. Kok, and W.-C. Siu, “A modified edge-directed interpolation for images,” *J. Electron. Imag.*, vol. 19, no. 1, pp. 013011-1–013011-20, Jan./Mar. 2010.
- [7] L. Zhang and X. Wu, “An edge-guided image interpolation algorithm via directional filtering and data fusion,” *IEEE Trans. Image Process.*, vol. 15, no. 8, pp. 2226–2238, Aug. 2006.
- [8] X. Zhang and X. Wu, “Image interpolation by adaptive 2-D autoregressive modeling and soft-decision estimation,” *IEEE Trans. Image Process.*, vol. 17, no. 6, pp. 887–896, Jun. 2008.
- [9] K.-W. Hung and W.-C. Siu, “Robust soft-decision interpolation using weighted least squares,” *IEEE Trans. Image Process.*, vol. 21, no. 3, pp. 1061–1069, Mar. 2012.
- [10] A. Marquina and S. J. Osher, “Image super-resolution by TV-regularization and Bregman iteration,” *J. Sci. Comput.*, vol. 37, no. 3, pp. 367–382, 2008.
- [11] S. Dai, M. Han, W. Xu, Y. Wu, Y. Gong, and A. K. Katsaggelos, “SoftCuts: A soft edge smoothness prior for color image super-resolution,” *IEEE Trans. Image Process.*, vol. 18, no. 5, pp. 969–981, May 2009.
- [12] J. Sun, J. Sun, Z. Xu, and H.-Y. Shum, “Gradient profile prior and its applications in image super-resolution and enhancement,” *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1529–1542, Jun. 2011.
- [13] H. Xu, G. Zhai, and X. Yang, “Single image super-resolution with detail enhancement based on local fractal analysis of gradient,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 10, pp. 1740–1754, Oct. 2013.
- [14] L. Wang, S. Xiang, G. Meng, H. Wu, and C. Pan, “Edge-directed single-image super-resolution via adaptive gradient magnitude self-interpolation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 8, pp. 1289–1299, Aug. 2013.
- [15] M. Protter, M. Elad, H. Takeda, and P. Milanfar, “Generalizing the nonlocal-means to super-resolution reconstruction,” *IEEE Trans. Image Process.*, vol. 18, no. 1, pp. 36–51, Jan. 2009.
- [16] K.-W. Hung and W.-C. Siu, “Single image super-resolution using iterative Wiener filter,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Kyoto, Japan, Mar. 2012, pp. 1269–1272.
- [17] K. Zhang, X. Gao, D. Tao, and X. Li, “Single image super-resolution with non-local means and steering kernel regression,” *IEEE Trans. Image Process.*, vol. 21, no. 11, pp. 4544–4556, Nov. 2012.
- [18] K. Zhang, X. Gao, D. Tao, and X. Li, “Single image super-resolution with multiscale similarity learning,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 10, pp. 1648–1659, Oct. 2013.
- [19] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael, “Learning low-level vision,” *Int. J. Comput. Vis.*, vol. 40, no. 1, pp. 25–47, 2000.

- [20] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Comput. Graph. Appl.*, vol. 22, no. 2, pp. 56–65, Mar./Apr. 2002.
- [21] H. Chang, D.-Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun./Jul. 2004, pp. 1–8.
- [22] W. Fan and D.-Y. Yeung, "Image hallucination using neighbor embedding over visual primitive manifolds," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2007, pp. 1–7.
- [23] T.-M. Chan, J. Zhang, J. Pu, and H. Huang, "Neighbor embedding based super-resolution algorithm through edge detection and feature selection," *Pattern Recognit. Lett.*, vol. 30, no. 5, pp. 494–502, Apr. 2009.
- [24] X. Gao, K. Zhang, D. Tao, and X. Li, "Image super-resolution with sparse neighbor embedding," *IEEE Trans. Image Process.*, vol. 21, no. 7, pp. 3194–3205, Jul. 2012.
- [25] M. Bevilacqua, A. Roumy, C. Guillemot, and M.-L. A. Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in *Proc. 23rd BMVC*, 2012, pp. 1–10.
- [26] K. Zhang, X. Gao, X. Li, and D. Tao, "Partially supervised neighbor embedding for example-based image super-resolution," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 2, pp. 230–239, Apr. 2011.
- [27] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2873, Nov. 2010.
- [28] J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. Huang, "Coupled dictionary training for image super-resolution," *IEEE Trans. Image Process.*, vol. 21, no. 8, pp. 3467–3478, Aug. 2012.
- [29] K. I. Kim and Y. Kwon, "Single-image super-resolution using sparse regression and natural image prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 6, pp. 1127–1133, Jun. 2010.
- [30] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1620–1630, Apr. 2013.
- [31] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Curves and Surfaces*. Berlin, Germany: Springer, 2012, pp. 711–730.
- [32] L. He, H. Qi, and R. Zaretzki, "Beta process joint dictionary learning for coupled feature spaces with application to single image super-resolution," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 345–352.
- [33] T. Peleg and M. Elad, "A statistical prediction model based on sparse representations for single image super-resolution," *IEEE Trans. Image Process.*, vol. 23, no. 6, pp. 2569–2582, Jun. 2014.
- [34] K. S. Ni and T. Q. Nguyen, "Image superresolution using support vector regression," *IEEE Trans. Image Process.*, vol. 16, no. 6, pp. 1596–1610, Jun. 2007.
- [35] C.-Y. Yang and M.-H. Yang, "Fast direct super-resolution by simple functions," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 561–568.
- [36] R. Timofte, V. De Smet, and L. Van Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 1920–1927.
- [37] R. Timofte, V. De Smet, and L. Van Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Proc. Asian Conf. Comput. Vis. (ACCV)*, 2014, pp. 1–15.
- [38] D. Dai, R. Timofte, and L. Van Gool, "Jointly optimized regressors for image super-resolution," *Eurographics*, vol. 34, no. 2, pp. 95–104, May 2015.
- [39] E. Pérez-Pellitero, J. Salvador, I. Torres-Xirau, J. Ruiz-Hidalgo, and B. Rosenhahn, "Fast super-resolution via dense local training and inverse regressor search," in *Proc. 12th Asian Conf. Comput. Vis. (ACCV)*, 2014, pp. 346–359.
- [40] K. Zhang, D. Tao, X. Gao, X. Li, and Z. Xiong, "Learning multiple linear mappings for efficient single image super-resolution," *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 846–861, Mar. 2015.
- [41] C. Dong, C. C. Loy, L. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. 13th Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 184–199.
- [42] Z. Cui, H. Chang, S. Shan, B. Zhong, and X. Chen, "Deep network cascade for image super-resolution," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 49–64.
- [43] J.-J. Huang and W.-C. Siu, "Practical application of random forests for super-resolution imaging," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Lisbon, Portugal, May 2015, pp. 2161–2164.
- [44] S. Schuler, C. Leistner, and H. Bischof, "Fast and accurate image upscaling with super-resolution forests," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3791–3799.
- [45] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *Proc. IEEE 12th Int. Conf. Comput. Vis. (ICCV)*, Sep./Oct. 2009, pp. 349–356.
- [46] G. Freedman and R. Fattal, "Image and video upscaling from local self-examples," *ACM Trans. Graph.*, vol. 30, no. 2, 2011, Art. ID 12.
- [47] H. He and W.-C. Siu, "Single image super-resolution using Gaussian process regression," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 449–456.
- [48] J. Yang, Z. Lin, and S. Cohen, "Fast image super-resolution based on in-place example regression," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 1059–1066.
- [49] L. Breiman, J. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Boca Raton, FL, USA: CRC Press, 1984.
- [50] J.-J. Huang, W.-C. Siu, and T.-R. Liu, "Fast image interpolation via random forests," *IEEE Trans. Image Process.*, vol. 24, no. 10, pp. 3232–3245, Oct. 2015.
- [51] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.



**Jun-Jie Huang** received the B.Eng. (Hons.) and M.Phil. degrees from The Hong Kong Polytechnic University, Hong Kong, in 2013 and 2015, respectively, under the supervision of Prof. W. C. Siu. He is currently pursuing the Ph.D. degree with Imperial College London, London, U.K.

His current research interests include image and video signal processing, image and video interpolation and super-resolution, image inpainting, and Internet-based signal processing.



**Wan-Chi Siu** (S'77–M'77–SM'90–F'12–Life–F'16) received the Ph.D. degree from the Imperial College of Science, Technology & Medicine, London, U.K., in 1984.

He joined The Hong Kong Polytechnic University, Hong Kong, in 1980. He has been the Chair Professor with the Department of Electronic and Information Engineering, since 1992. He was an Independent Non-Executive Director of Teleeve Holding Ltd., Hong Kong, from 2000 to 2015, which is a listed video surveillance company. He is

currently the Director of the Centre for Signal Processing, and was the Head (EIE) and, subsequently, the Dean of the Engineering Faculty with The Hong Kong Polytechnic University from 1994 to 2002. He is an Expert in Digital Signal Processing, specializing in fast algorithms, video coding, and pattern recognition. He has authored over 490 research papers, and holds eight recent patents.

Prof. Siu is a fellow of the Institution of Engineering and Technology and the Hong Kong Institution of Engineers. He received many awards, such as the Distinguished Presenter Award, the Best Researcher Award, and the IEEE Third Millennium Medal. He was the Vice President of the IEEE Signal Processing Society from 2012 to 2014. He has recently been elected as the President-Elect of the Asia-Pacific Signal and Information Processing Association from 2015 to 2016. He was a Guest Editor, an Associate Editor, and a member of the Editorial Boards of a number of journals, including the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS and the IEEE TRANSACTIONS ON IMAGE PROCESSING from 2010 to 2012. He is the Subject Editor (in charge of Image Processing) of Electronics Letters from 2015–2018, an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY from 2015 to 2017, in addition to other journals. He is a Popular Lecturing Staff Member within the university, while he has been a Keynote Speaker of over 12 international/national conferences in the recent ten years outside the university. He is the General Chair and Technical Program Chair of several prestigious IEEE Society's flagship international conferences (including the International Conference on Image Processing in 2010, the International Conference on Acoustics, Speech, and Signal Processing in 2003, and the International Symposium on Circuits and Systems in 1997). In 1992/1993, he chaired the first Engineering and Information Technology Panel of the Research Assessment Exercise and initiated some milestone basic quality measures to assess the research quality of academia in universities, which had a long-term impact on the development of quality research in Hong Kong.