Piecewise Linear Regression-Based Single Image Super-Resolution via Hadamard Transform

Outline

- Extracting training data
- Image feature representation
- Clustering training data
- Computing Mapping models
- Testing
- Experimental results
- Conclusion

Extracting training data

LR (low-resolution)
HR (high-resolution)

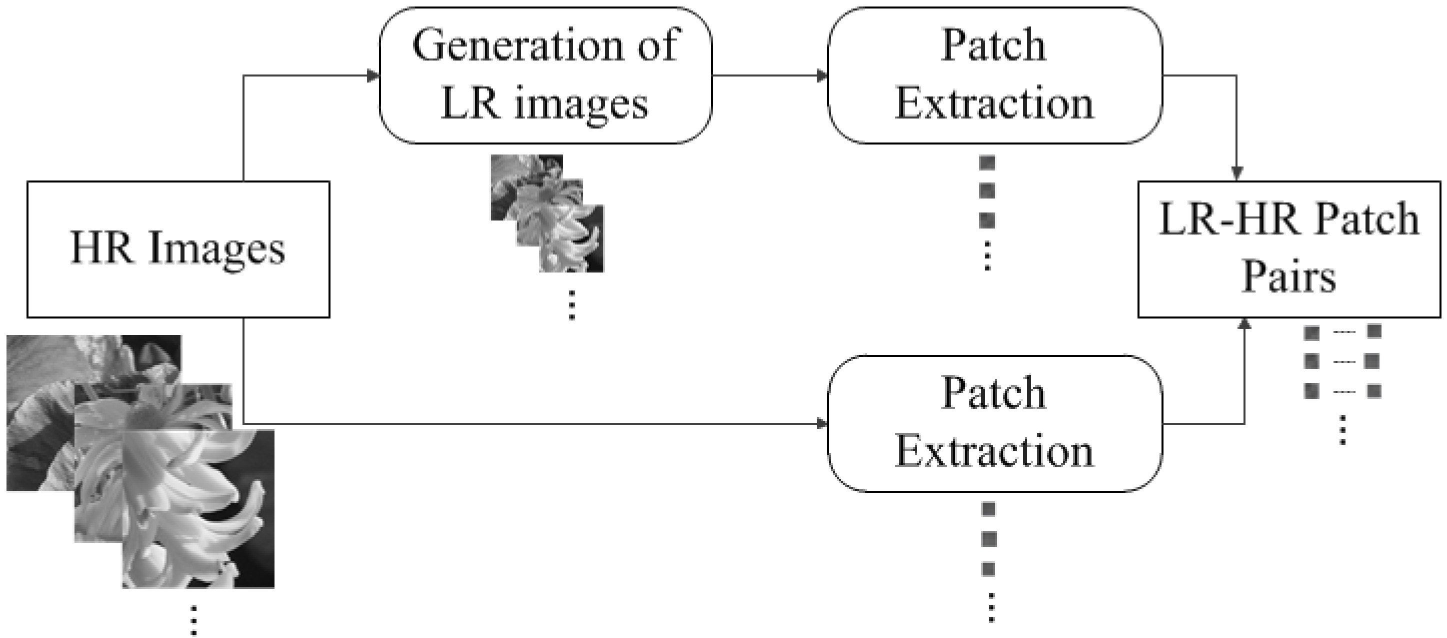


Fig.1. Extraction of training data

An LR-HR patch pair contains an LR image patch and its corresponding HR image patch.

Extracting training data

Now we extract LR-HR patch pairs from an LR-HR image pair ($imageL$, $imageH$).
One pad is added around $imageL$, which is one pixel in width, to get $imagepad$.
$sz(1) = width(imagepad)$, $sz(2) = height(imagepad)$

The LR image patch :

$$imagepad(i{:}\, i + 3, j{:}\, j + 3)$$

The corresponding HR image patch :

$$imageH((i-1) * scale + offset + 1{:}\, i * scale + offset, (j-1) * scale + offset + 1{:}\, j * scale + offset)$$

where $2 \le i \le sz(1) - 4, 2 \le j \le sz(2) - 4$, $scale$ is the upscaling factor and $offset = \lfloor scale/2 \rfloor$.

Each image patch is represented by a row vector. All the vectorized LR image patches are stacked to form a matrix. So does the vectorized HR image patches.

Image feature representation

The Hadamard matrix is used to extract image features, its formula is as follow:

$$Q_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$Q_{2^{k+1}} = \begin{bmatrix} Q_{2^k} & Q_{2^k} \\ Q_{2^k} & -Q_{2^k} \end{bmatrix}$$

A 16-order Hadamard matrix Q_{16} is used in our paper. The first column of it is all 1. We delete the first column of Q_{16} to get a new matrix Q_{15}.

Image feature representation

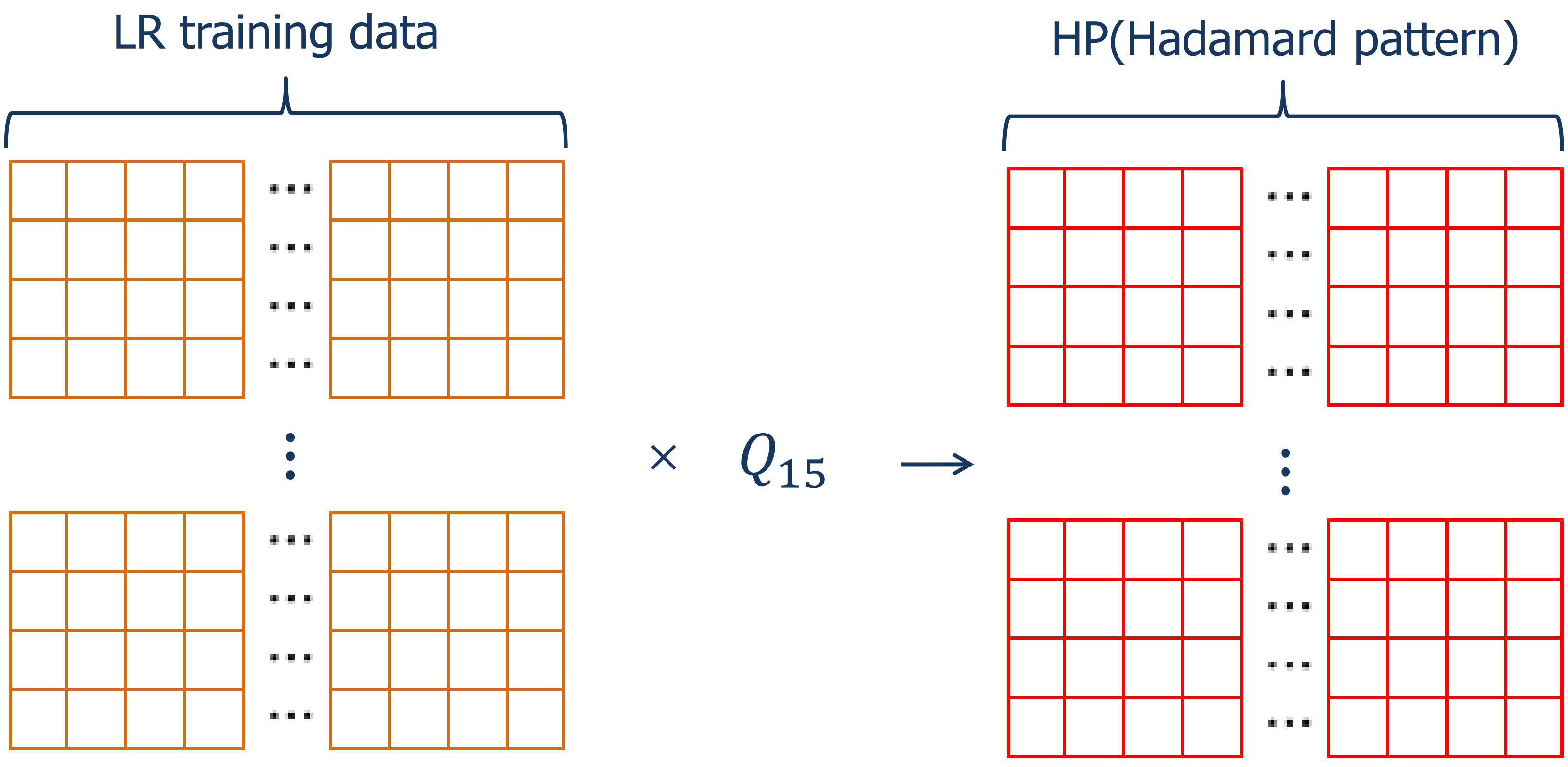


Fig.2. Computing the Hadamard patterns of LR training data

Clustering training data

Each column of Q_{15} is equivalent to a convolution filter.

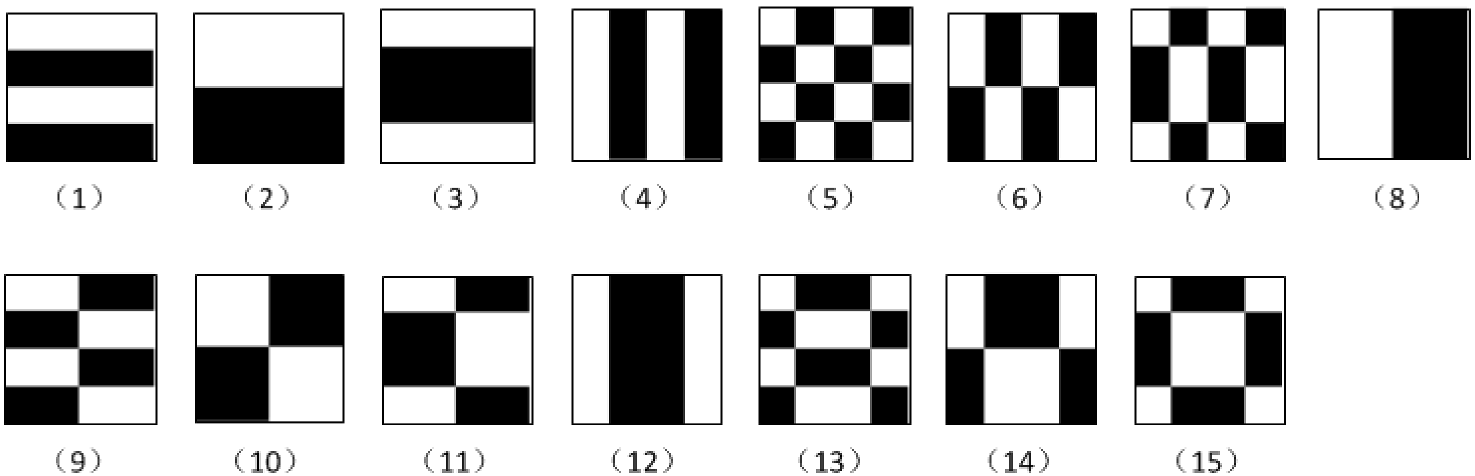
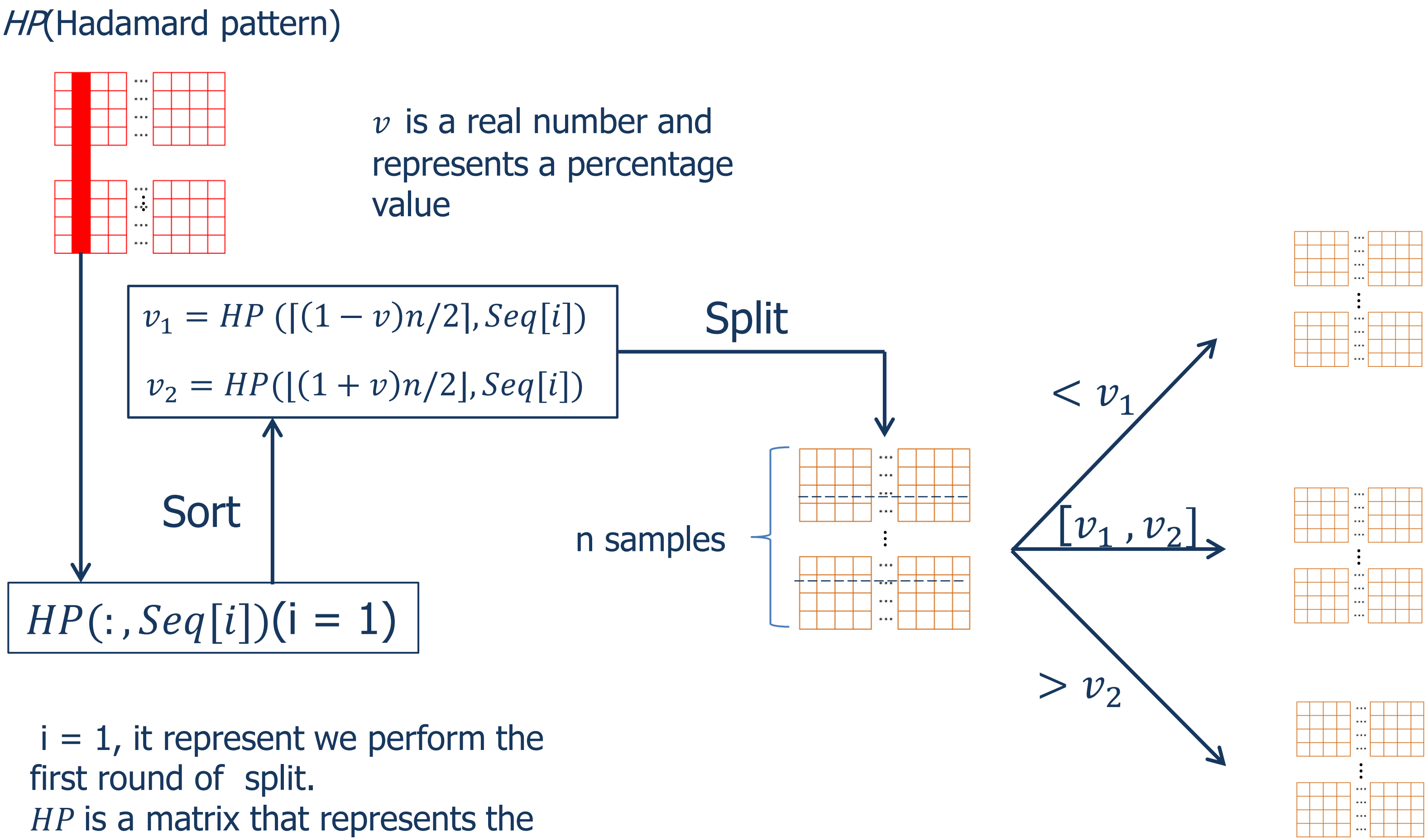


Fig.3. The visualization of Q_{15}

Set the sequence :

Seq = [2 8 3 12 10 1 4 11 14 6 9 15 7 13 5].

Clustering training data



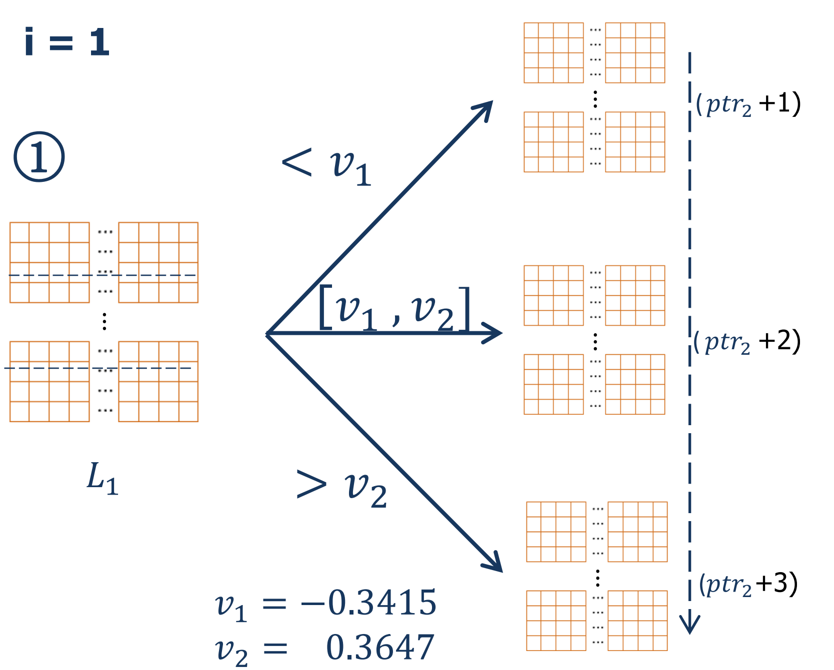
i = 1, it represent we perform the first round of split.
HP is a matrix that represents the generated Hadamard patterns.
v_1 and v_2 are the thresholds we learned and they are real numbers.

Fig.4. The illustration of one split

Clustering training data

ptr_0: the current line number of the parameter matrix, its initial value is 1.	ptr_1: the current line number of the decision tree, its initial value is 1.
ptr_2: the front line number of the decision tree, its initial value is 1.	lr, hr: store the training data that have arrived at leaf nodes. They consist of matrices.

i = 1



②

	i	ptr_2+1	ptr_2+2	ptr_2+3	v_1	v_2
ptr_1-> 1	1	2	3	4	-0.3415	0.3647
2						
3						
4						
5						
6						
7						
8						

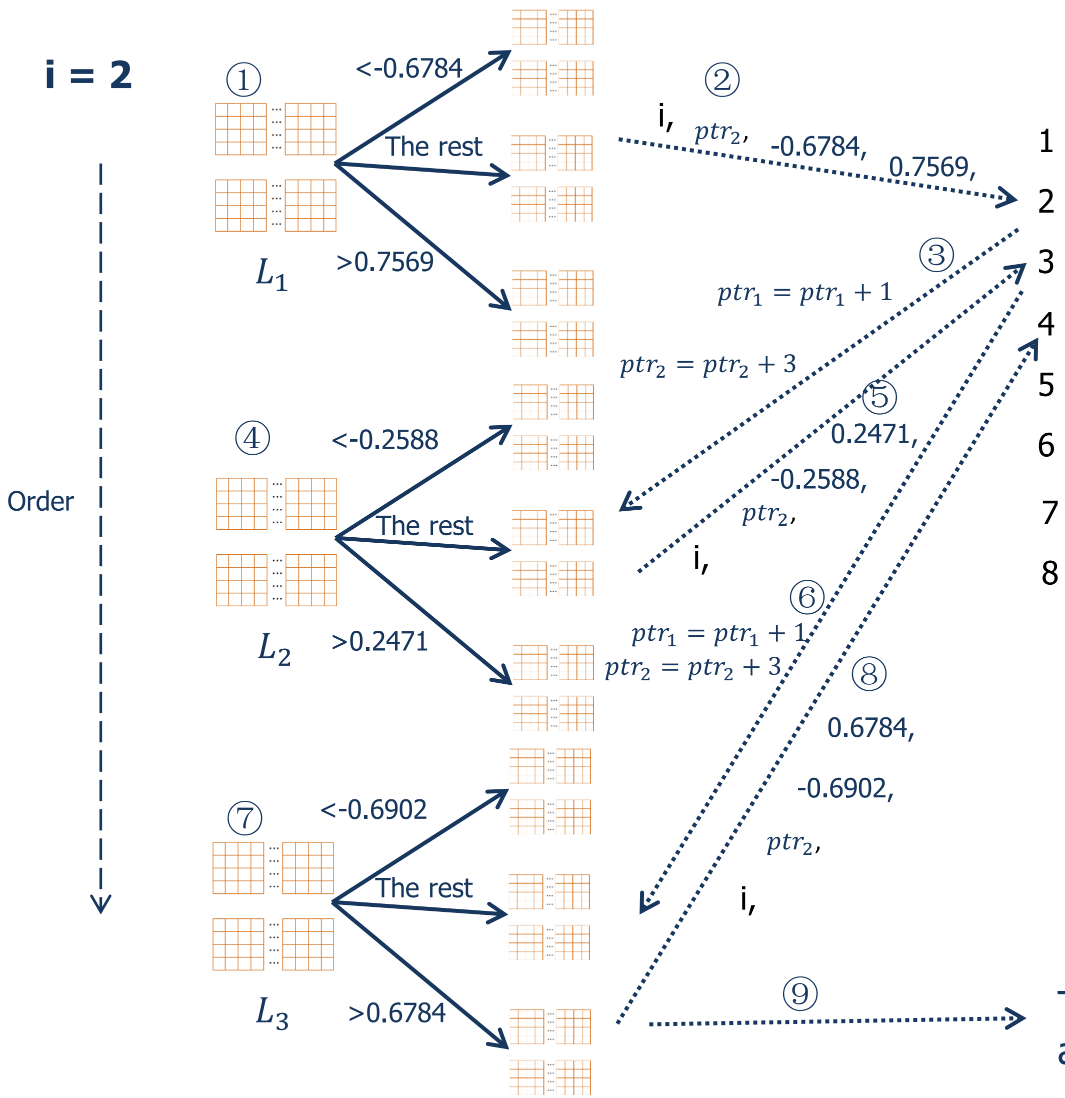
⋮

The matrix of the SR decision tree

③ Then $ptr_1 = ptr_1 + 1$, $ptr_2 = ptr_2 + 3$ and delete L_1.

Fig.5. The illustration of constructing a SR decision tree

Clustering training data



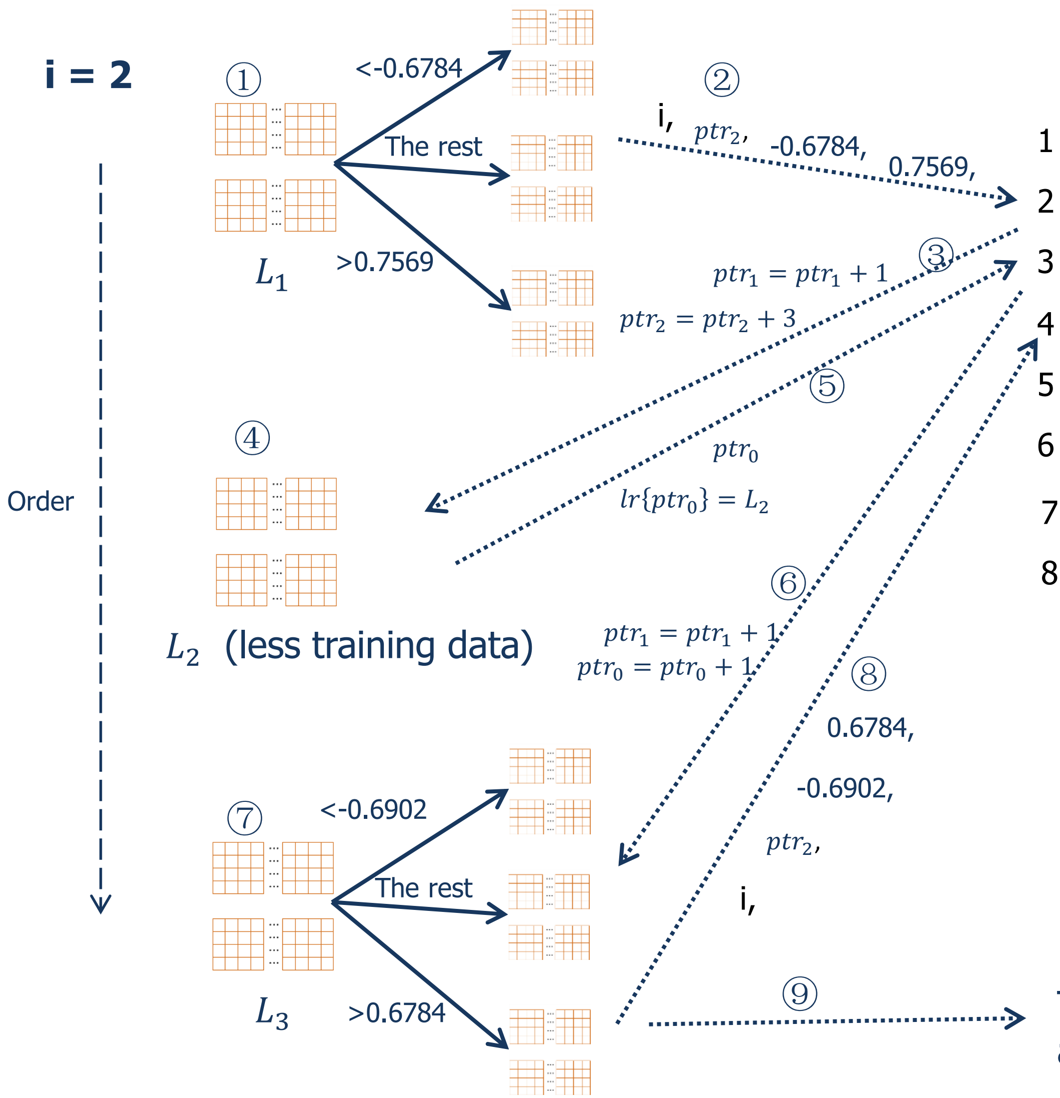
	i	ptr_2+1	ptr_2+2	ptr_2+3	v_1	v_2
1	1	2	3	4	-0.3415	0.3647
2	2	5	6	7	-0.6784	0.7569
3	2	8	9	10	-0.2588	0.2471
4	2	11	12	13	-0.6902	0.6784
5						
6						
7						
8						

⋮

Then $ptr_1 = ptr_1 + 1$, $ptr_2 = ptr_2 + 3$
and delete L_1, L_2 and L_3.

Fig.6. The illustration of constructing a SR decision tree

Another case



	i	ptr_2+1	ptr_2+2	ptr_2+3	v_1	v_2
1	1	2	3	4	-0.3415	0.3647
2	2	5	6	7	-0.6784	0.7569
3	2	0	1	0	0	0
4	2	8	9	10	-0.6902	0.6784
5						
6						
7						
8						

⋮

Then $ptr_1 = ptr_1 + 1$, $ptr_2 = ptr_2 + 3$ and delete L_1, L_2 and L_3.

Fig.7. The illustration of constructing a SR decision tree

Computing Mapping models

The corresponding high-resolution training data are split in the same way as the low-resolution training data are split.

for each element in lr **do**

$$M_q = least\_square(lr_q, hr_q);$$

endfor

where M_q is the mapping model of the q^{th} leaf node, which is a coefficient matrix. The constraint is that the sum of each column of M_q is 1.

Testing

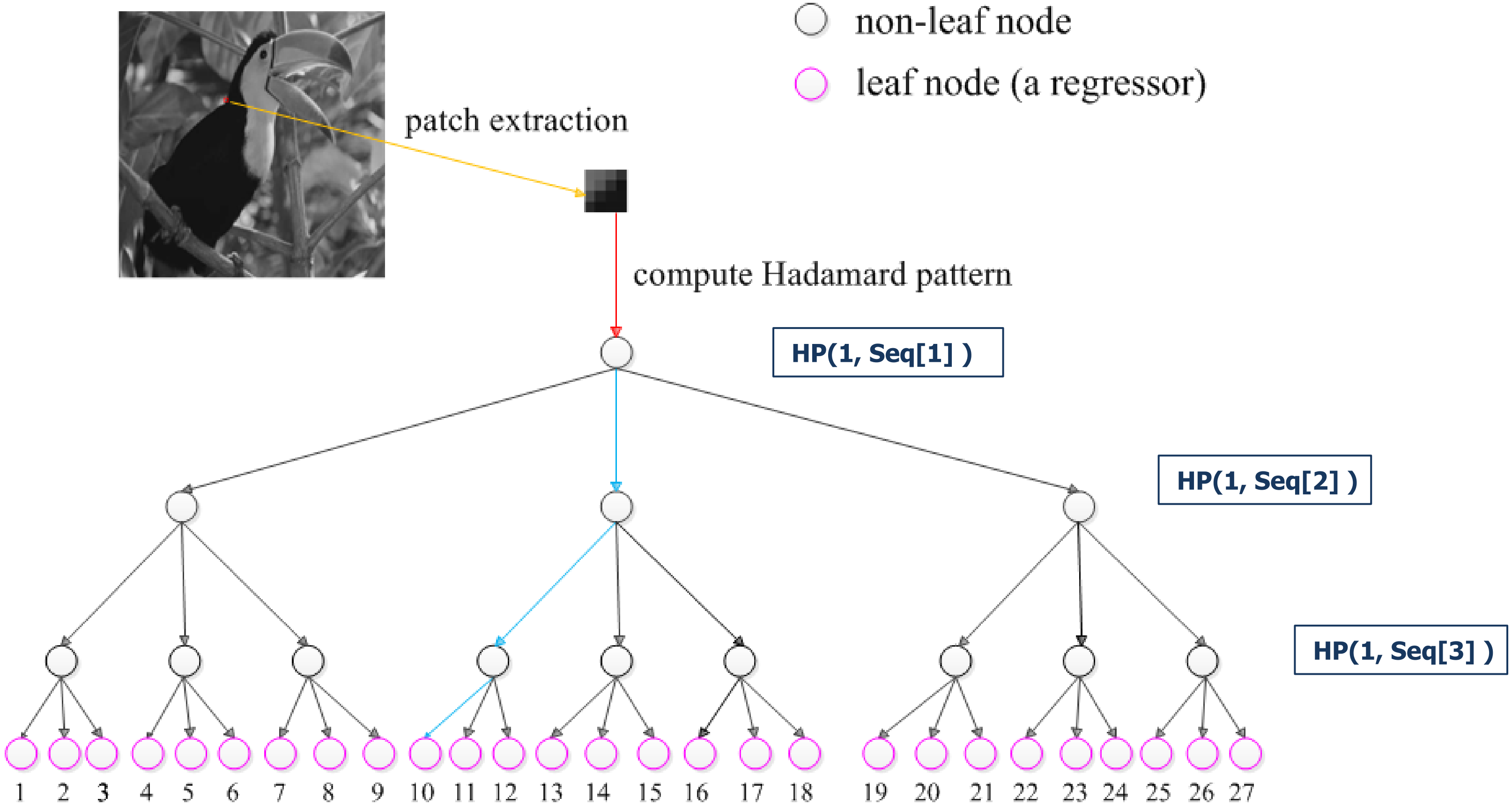


Fig.8. Super-resolution scheme

In the testing phase, the testing images are different from the training images. Here HP is the Hadamard patterns of one LR image patch and is a row vector. HP(1,Seq[1]) represents a element of HP and it is a real number.

Testing

Seq = [2 8 3 12 10 1 4 11 14 6 9 15 7 13 5].

HP=[-0.8824, 0.2316, -0.6314, -0.5765,
0.1686, 0.0196, 0.0588, -0.5843,
-0.0118, 0.3098, -0.5137, -2.0431,
0.0980, -0.2235, 0.8667];

HP(1, Seq[1]) = **0.2316**, -0.3415<**0.2316**<0.3647
So the middle child node (**3**) is selected. Go to the third row of the learned decision tree.

HP(1, Seq[2]) = **-0.5843**, **-0.5843**< -0.2588. So the left child node (**8**) is selected. Go to the eighth row of the learned decision tree.

HP(1, Seq[3]) = **-0.6314**, **-0.6314**<-0.1373. So the left child node (**23**) is selected. Go to the 23^{th} row of the Learned decision tree.

1	1	2	3	4	-0.3415	0.3647
2	2	5	6	7	-0.6784	0.7569
3	2	8	9	10	-0.2588	0.2471
4	2	11	12	13	-0.6902	0.6784
5	3	14	15	16	-0.3647	0.3608
6	3	17	18	19	-0.3882	0.4039
7	3	20	21	22	-0.3686	0.3725
8	3	23	24	25	-0.1373	0.1333
9	3	26	27	28	-0.0745	0.0745
10	3	29	30	31	-0.1412	0.1412
11	3	32	33	34	-0.3451	0.3412
12	3	35	36	37	-0.3843	0.3804
13	3	38	39	40	-0.3216	0.3216
14	4	0	1	0	0	0
15	4	0	2	0	0	0
16	4	0	3	0	0	0

Testing

17	4	0	4	0	0	0
18	4	0	5	0	0	0
19	4	0	6	0	0	0
20	4	0	7	0	0	0
21	4	0	8	0	0	0
22	4	0	9	0	0	0
23	4	0	10	0	0	0
24	4	0	11	0	0	0
25	4	0	12	0	0	0
26	4	0	13	0	0	0
27	4	0	14	0	0	0
28	4	0	15	0	0	0
29	4	0	16	0	0	0
30	4	0	17	0	0	0
31	4	0	18	0	0	0
32	4	0	19	0	0	0

33	33	0	20	0	0	0
34	34	0	21	0	0	0
35	35	0	22	0	0	0
36	36	0	23	0	0	0
37	37	0	24	0	0	0
38	38	0	25	0	0	0
39	39	0	26	0	0	0
40	40	0	27	0	0	0

So M_{10} is used to generate target high-resolution patch.

Testing

Fig.9. The position relationship between an input low-resolution patch and its predicted high-resolution patch

Experimental results

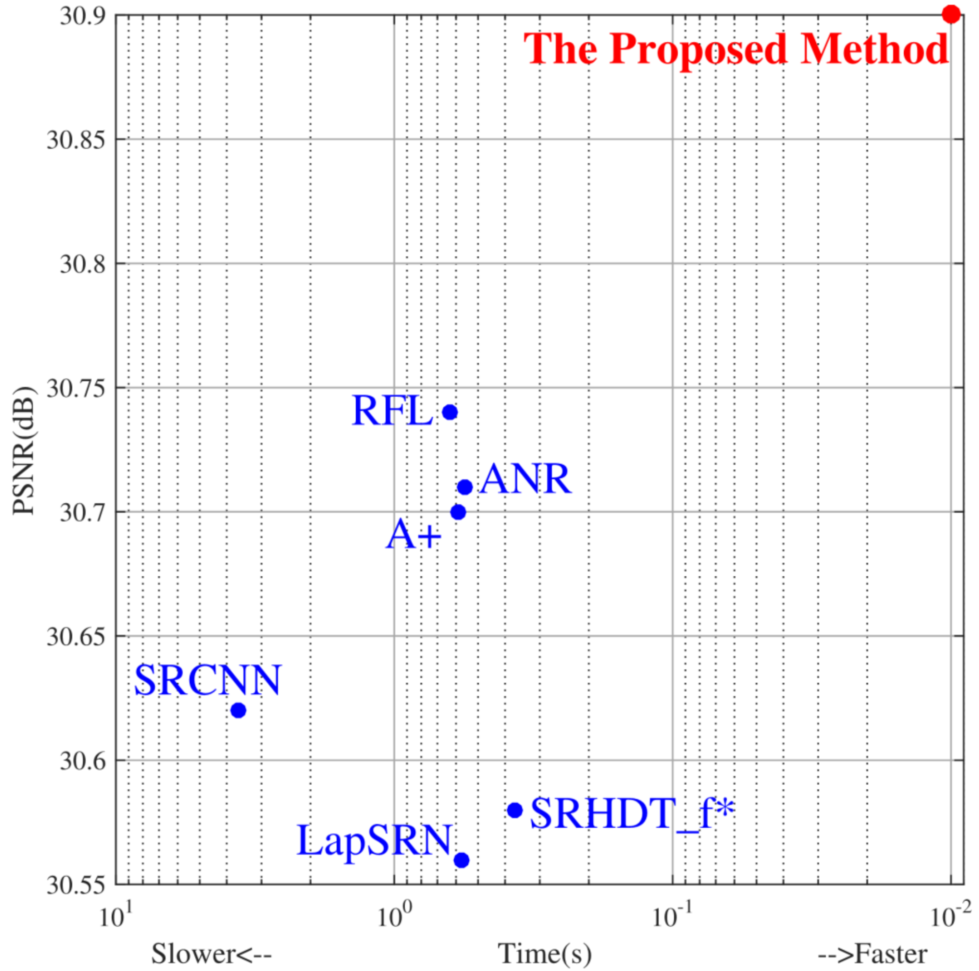


Fig.10. Speed and accuracy trade-off. The results are evaluted on Set5 with upscaling factor 2.

Experimental results

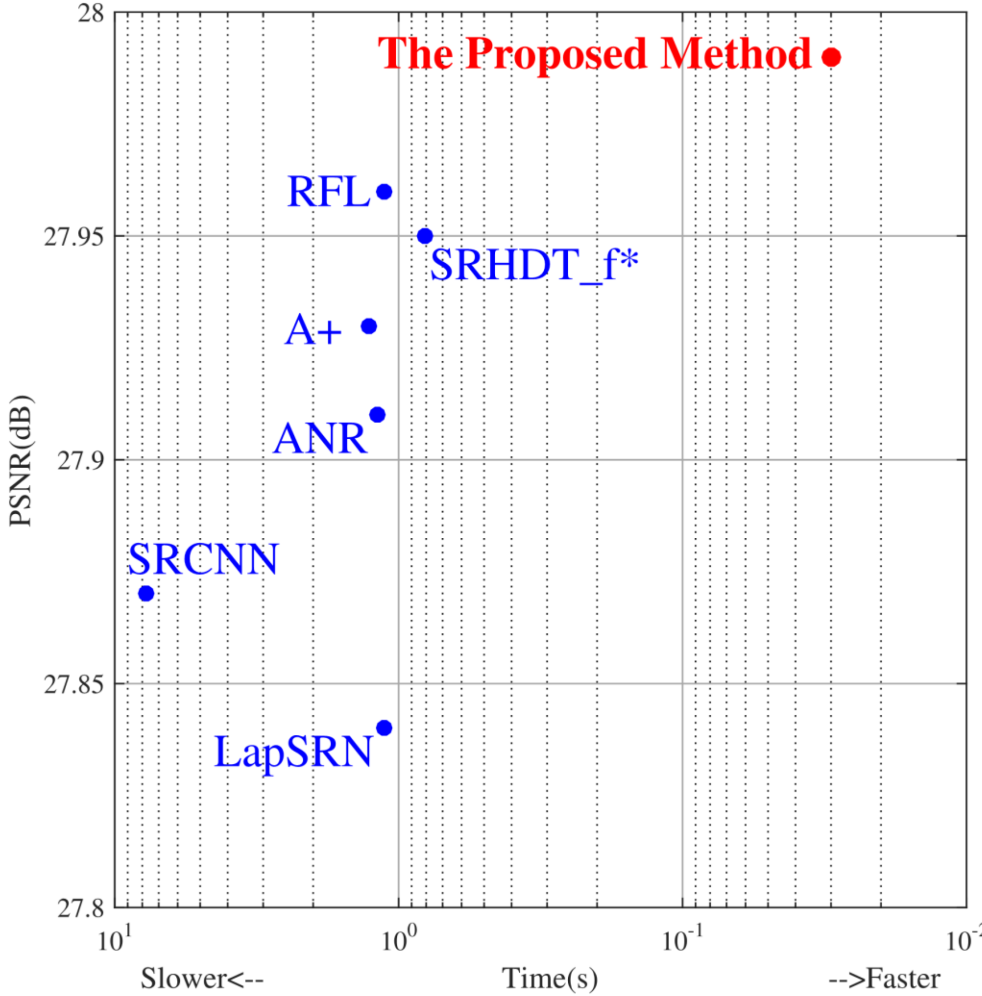
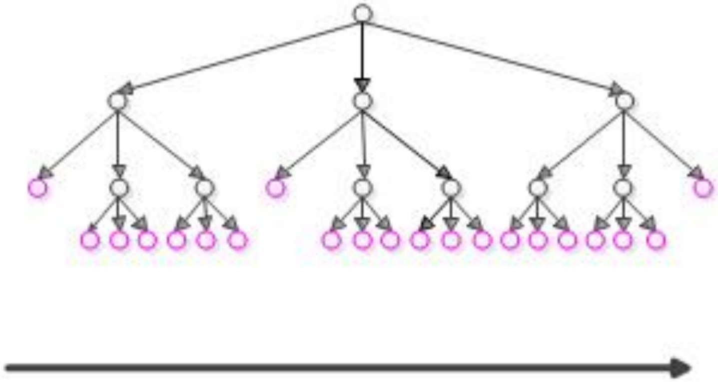


Fig.11. Speed and accuracy trade-off. The results are evaluated on Set14 with upscaling factor 2.

Conclusion



Single Image Super-Resolution

- Super-Resolution Decision Tree

Thank You!