# A NOVEL SUPER-RESOLUTION METHOD BASED ON PATCH RECONSTRUCTION WITH SIMK CLUSTERING AND NONLINEAR MAPPING

*Peiqi Duan, Anlong Ming, Xuesong Zhang, Xuejing Kang*[*]

Beijing University of Posts and Telecommunications. Beijing 100876, China

## ABSTRACT

In this paper, we propose a patch-wise super-resolution (SR) method that combines an external-sample classification tree and a nonlinear-mapping learning stage to simultaneously guarantees reconstruction quality and speed at the stage of patch representation and mapping. We use the low-resolution (LR) to high-resolution (HR) mapping kernel of each patch-pair sample (called SIMK) to complete classification by binary tree branching and provide reasonable training sets for mapping-learning. Then a high accuracy but low cost lightweight network is learned for each tree node to choose the reasonable branch path for the testing LR patches. In the mapping-learning stage, the nonlinear mapping for each class is represented as a full-connected network, which provides satisfying generalization ability for LR patch reconstruction. Comparing with state-of-the-art methods, our approach achieves real-time (>24fps) SR of realistic vision and high quality for different upscaling factors.

***Index Terms—*** Image super-resolution, external learning, classification, nonlinear regression

## 1. INTRODUCTION

Single image super-resolution (SR) is a significant basic technique in the fields of computer vision. In general, SR problem is supposed to be a linear degradation model [1], the low-resolution (LR) image is generated from high-resolution (HR) one suffering a blurring process and a down-sampling process. SR methods aim to achieve the inverse process of the degradation model, whereas this is well known as an ill-posed inverse problem [1], because the model is nonlinear and there may not exist a unique HR image generated from one input LR image. Various SR methods have been proposed in recent years to improve image reconstruction quality, which can be categorized as interpolation-based [2], reconstruction-based [3, 4] and learning-based methods [5, 6].

Currently, state-of-the-art SR reconstruction qualities are almost obtained by external-example learning-based methods [7, 8], which endlessly explore more appropriate patch representations and mapping functions to accurately express the LR-to-HR relationship by external samples. Generally, patch representation and mapping stages are the important parts that affect quality and speed of SR [8]. Patch representation based on sparse-coding [9, 10] usually learns a joint dictionary of HR-to-LR to synthesize generative HR patches with sparse coefficients. As improved methods, ANR [7] and A+ [11] use the least-squares to combine the joint dictionary into a mapping matrix. Classification-based patch representation [12, 13] uses gradient features of LR patch to classify LR-HR patch-pair samples and learn a linear mapping for each class to generate desired HR patches from the testing LR patches. Those sparse-coding and classification-based patch representations can quickly characterize samples, but they only consider the features of LR patch and ignore the LR-to-HR mapping relationship of each sample, which will cause inaccurate patch representation. For the mapping stage, since SR is an ill-posed inverse problem, linear-mapping usually lead to higher regression error and affect the restoration quality. As a hotspot, CNN-based methods [8, 14] avoid this problem and achieve amazing performance by directly creating end-to-end networks between LR and HR images. The multi-layer network can store a widely compatible mapping relationship but a vast number of network parameters are hard to converge and will lead to the runtime increasing. Besides, the large receptive field (i.e. patch size) that matches with the multi-layer structure may ignore the feature extraction of tiny local details, which will lead to the cartoon effect and unnatural visual of reconstructed HR images.

In this paper, we propose a real-time and high quality SR method that combines a sample classification-tree based on Sample Individual Mapping-Kernel (SIMK) and a mapping-learning stage based on nonlinear regression, which simultaneously guarantees the reconstruction quality and speed from patch-representation and mapping-learning stages. In the patch-representation stage, we use the SIMK that is the LR-to-HR mapping kernel of each sample, to classify samples and obtain reasonable sample sets for mapping-learning, which also helps ensure both global and detail regions reconstruction by setting a small patch size. In the mapping-learning stage, one nonlinear mapping is represented as a lightweight network for each class, which provides satisfying generalization ability for LR patch reconstruction.

[*]corresponding author. This work was supported in part by the National Natural Science Foundation of China (61701036, 61871055), Fundamental Research Funds for the Central Universities (2018RC54).

## 2. PRELIMINARY

Our method builds on theories from mapping-learning-based SR algorithms. Generally, single image degradation process can be represented by the following equation:

$$Y = \mathbf{DB}X + n \tag{1}$$

where $Y$ is the LR image generated from HR image $X$ suffering a blurring process with the operator $\mathbf{B}$, a down-sampling process with the operator $\mathbf{D}$ and an additive noise $n$. SR problem can be formulated as a Maximum a posteriori estimation:

$$\begin{aligned} \hat{X} &= \arg\max_{X} f(X, \mathbf{D}, \mathbf{B}, \sigma^2 \,|Y\,) \\ &= \arg\min_{X} L(Y \,\big|X, \mathbf{D}, \mathbf{B}, \sigma^2\,) + L(X) \\ &= \arg\min_{X} \frac{1}{2\sigma^2} \|Y - \mathbf{DB}X\|_2^2 + L(X) \end{aligned} \tag{2}$$

where $f(.)$ is the prior distribution, $L(.) = -log(f(.))$ and we assume the noise term of equation (1) is follow Gaussian distribution with parameter $\sigma$. $L(X)$ is the prior constraint which we will obtain from external LR-HR patch-pair samples by mapping-learning.

## 3. THE PROPOSED SUPER-RESOLUTION METHOD

In this section, we propose a real-time and high-quality SR method with the prior constraint obtainment by sample classification and mapping learning. We classify external samples based on SIMK and learning one LR-to-HR nonlinear mapping for each class, which aims to solve the optimization problem similar to [12, 13, 15]:

$$\min_{c, M_c} \sum_{c=1}^{k} \sum_{l_i \in c} \|h_i - M_c l_i\|_2^2, \tag{3}$$

where $l_i/h_i$ are LR/HR patch vectors collected from training set, $c$ is the class label and $M_c$ means the desired nonlinear mapping of class $c$. Then the equation (2) can be represented as follows to generate an SR image $\hat{X}$:

$$\hat{X} = \arg\min_{X} \|Y - \mathbf{DB}X\|_2^2 + \lambda \sum_{i} \|Q_i X - M_{c_i} y_i\|_2^2 \tag{4}$$

where $y_i$ is an LR patch collected from the testing LR image $Y$, and $Q_i$ is the matrix extracting the corresponding local patch. $c_i$ means the class label of $y_i$ and $M_{c_i}$ is the corresponding mapping. Therefore the main aim of the training process is to achieve appropriate sample classification and learn reasonable mappings to optimize equation (3).

As Fig. 1 shows, in the training process, we use a binary-tree to classify samples based on SIMK-clustering. Then a branch Neural Network (b-NN) is learned for each non-leaf node and an nonlinear mapping Neural Network (m-NN) is learned for each class (leaf node). In the testing process, LR patches search branch paths and match with different classes by b-NNs and complete upsampling with the corresponding m-NN. Then the generated HR image is further optimized by equation (4). Next we will introduce our method in detail.

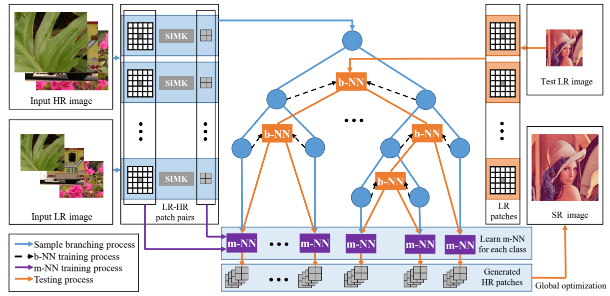


**Fig. 1**. The flow chart of our method.

### 3.1. Sample Branch Tree Based on SIMK-Clustering

Without performing bicubic initial upsampling, our method directly uses the neighborhood patch to amplify each pixel of LR image, which effectively improves the run speed similar to [12]. Assume that the upsampling factor is $s$, we collect overlapping LR patches $l \in \mathbb{R}^{n \times n}$ from LR images and correspond HR patches $h \in \mathbb{R}^{s \times s}$ of HR images directly.

Different from the methods which distinguish features of LR patch to classify LR-HR patch pairs, our method uses SIMK, which is the LR-to-HR mapping kernel of each sample, as the classification basis. Consider a vectorized sample $p_k : \{l_k, h_k\}$, there exists a mapping relationship as $h_k = m_k l_k$ and we define the matrix $m_k \in \mathbb{R}^{s^2 \times n^2}$ as the SIMK of the sample $p_k$, which is calculated as $m_k = h_k l_k^{-1}$. Then the optimization equation (3) can be further expressed as:

$$\min_{c, M_c} \sum_{c=1}^{k} \sum_{m_i \in c} \|(m_i - M_c) l_i\|_2^2. \tag{5}$$

This equation clearly shows that for LR-HR patch-pairs in one class $c$, the aggregation degree of set $\{m_i \in c\}$ is more directly affecting the accuracy of the regression result $M_c$ than the aggregation degree of set $\{l_i \in c\}$. To quantitatively compare two sample classification methods, we randomly extract 0.4 million samples into 200 groups to compare the difference of regression errors. For each group, we perform $l$-based and $m$-based clustering respectively and extract samples $\{p_k^l\}$ and $\{p_k^m\}$ (k = 1,..., 500), which are closest to clustering centers. Then we learn mappings as [12], obtain regression errors by calculating Mean Square Error and show the histograms in Fig. 2. High regression accuracy corresponds small error, so it is obvious that the SIMK-based classification we use is more

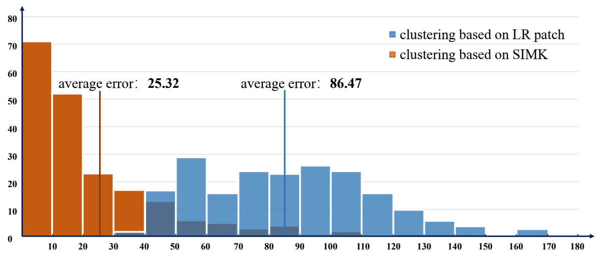


**Fig. 2**. The comparison of two different sample classifications. Blue (orange) bars represent the histogram of regression errors based on LR patch (SIMK) classification.

accurate than the LR patch-based method.

We utilize the binary decision tree to perform SIMK-based sample classification and aim to approach the idealized optimal result of the equation (2). LR-HR patch-pairs $\{p_i|p_i = (l_i, h_i, m_i)\}$ collected from training set are branched into two child nodes on each non-leaf by k-means clustering of $\{m_i\}$, and finally classified into leaf nodes. Considering the high run cost and the effect of feature correlation, we use PCA to reduce the dimension of $m$ before the clustering. Fig. 3 shows the clustering process of one non-leaf node, which includes sample branch process, b-NN and m-NN learning process, and validation process.

### 3.2. Branch Network Learning for The Testing LR Patch and Mapping Learning Based on Nonlinear Regression

One important problem of our method is how to find the right branch path for LR patches in the testing process, because there is no SIMK during this stage. Duan *et al.* [15] uses Naive Bayes Classifier to learn probabilistic models for LR patches, but the accuracy only reaches 70%. To further improve the classification accuracy, as Fig. 3 shows, we learn a full-connected neural network b-NN with one hidden layer for each non-leaf node, where the corresponding branch labels $\{lab_i|lab \in \{-1, 1\}\}$ are obtained by clustering $\{m_i\}$. The cost function is expressed as follows:

$$L_{b-NN} = \min_{W,b} \sum_i \|lab_i - f(W' \cdot f(Wl_i + b) + b')\|_2^2 + \lambda(\|W\|_F^2 + \|W'\|_F^2), \tag{6}$$

where $W/W'$ denotes the connection weight matrix of input/hidden layers, $b/b'$ denotes the corresponding bias values, and $f(.)$ is a group of non-linear sigmoid functions. Then network parameters can be updated iteratively by BP algorithm.

External LR-HR patch pairs are collected from natural images and mapping relationships between LR and HR patches exist uncertainty, so linear regression is difficult to fit them well. Therefore, we use a BP network to represent the nonlinear mapping relationship and achieve the generalization of

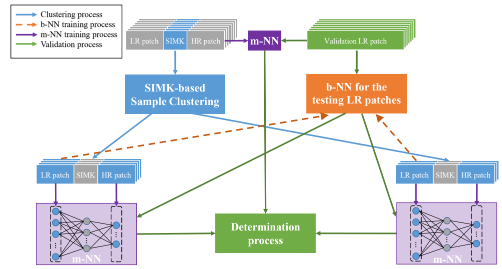


**Fig. 3**. The flow chart of non-leaf node branching process.

the mapping. As Fig. 3 shows, similar to b-NN, we also use a full-connected network to learn nonlinear mapping m-NN for each node. These m-NNs memorize the inherent regularity of extracted datas into the network connection weight, which has outstanding approximation ability and low complexity. The cost function of 2-layer m-NN is expressed as follows:

$$L_{m-NN} = \min_{W,b} \sum_i \|h_i - f(W' \cdot f(Wl_i + b) + b')\|_2^2 + \lambda(\|W\|_F^2 + \|W'\|_F^2). \tag{7}$$

The m-NN aims to represent the intrinsic mapping relationship between the patch-level input and output, so one or two hidden layer can effectively learn the mapping to complete LR patch reconstruction with high speed. In addition, benefit from SIMK-based classification, which is essentially a feature extraction stage, we can set a small LR patch size so that m-NN will focus on tiny details and reconstruct them well.

### 3.3. Decision Scheme for Branching Nodes

As we described above, the purpose of node branching is to classify samples with similar mapping relationships into the same child node, thereby guaranteeing the generalization ability of m-NNs. To ensure the validity of node branching, we design a validity check to decide whether to perform branching for each node as shown in Fig. 3. We collect validation set $S = \{vp_i|vp_i = (vl_i, vh_i)\}$ from BSD500 [16] and complete selection of optimal result as follows,

$$\min\Big(\sum_{vl_i \in S_1} \|vh_i - F_1(vl_i)\|_2^2 + \sum_{vl_j \in S_2} \|vh_j - F_2(vl_j)\|_2^2, \sum_{vl_k \in S_0} \|vh_k - F_0(vl_k)\|_2^2\Big), \tag{8}$$

where $vl/vh$ are the LR/HR patches of father node which are classified from the validation set, $F_0(.)$ denotes the learned m-NN of father node and $F_1(.)/F_2(.)$ denote the m-NN of right/left child nodes. In this process, We use m-NNs of the father-node and two branched child-nodes to reconstruct LR patches of the classified samples respectively, then calculate the squared errors of each layer and confirm whether the branching of father node is valid.

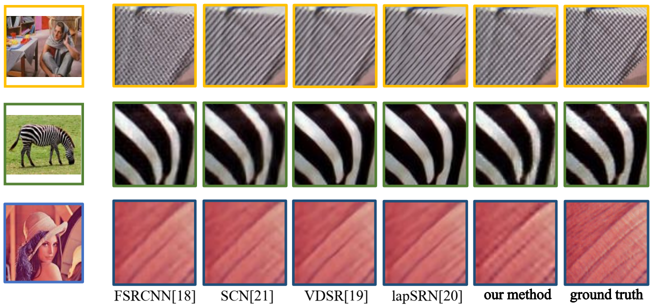


**Fig. 4**. Visual comparison with other methods for $\times 4$ amplification. **(Zoom in for best view)**

By iteratively executing the proposed node-branching process, we obtain the desirable decision tree and m-NN of each leaf node. In the testing process, patches of LR image select the path and match leaf nodes through the learned b-NNs, then generate HR patches by the corresponding m-NNs and reconstruct the whole image. Finally, to further improve the quality by global constraint, the HR image is optimized by equation (4), where the closed-form solution is obtained similar to the method [17]:

$$\hat{X} = \mathcal{F}^{-1}\left(\frac{\mathcal{F}(\mathbf{B}^{\mathrm{T}}\mathbf{S}^{\mathrm{T}}Y) + \lambda\mathcal{F}(\sum_i (M_i l_i)^{\mathrm{T}} Q_i)}{\mathcal{F}(\mathbf{B}^{\mathrm{T}}\mathbf{S}^{\mathrm{T}}\mathbf{S}\mathbf{B}) + \lambda\mathcal{F}(\sum_i {Q_i}^{\mathrm{T}} Q_i)}\right) \quad (9)$$

$\mathcal{F}(.)$ denotes the Fourier transform and $M_i l_i$ can be considered as $i$-th gerenated HR patch.

## 4. EXPERIMENT

In our experiments, we use 728 images as the training set like [13]. Methods are tested to evaluate the performance of upscaling factors 2,3 and 4. LR images are created by downsizing HR images with bicubic sampling and Gaussian blur. Both SR methods are only applied to and compared on the luma channel. We collect LR patches with size $5 \times 5$ from the training set. For different upscaling factor s, the corresponding HR patch size is $s \times s$.

Our method uses BP algorithm to train several b-NNs and m-NNs based on fully-connected neural networks, where proper network parameter set can balance performance and runtime. After experimental analysis, we set 2-layer b-NNs as {25,15,1} networks and set m-NN as {25,50,15,$s^2$} ones, which can achieve desired accuracy as well as fast speed. The proposed method is compared with other SR algorithms for both objective and visual quality assessments. Table 1 shows the numerical results, where 'our-s' denotes the results without global constraint. It shows our method achieves the best results with the highest PSNR and SSIM values. Fig. 4 shows the comparison of the visual quality on upscaling factor 4. Comparing with others which successfully reconstruct clearer textures and smoother edge lines but exist over-sharpen effects, our method better reconstructs the local noise and edge aliasing that usually exist in the texture region, making the visual closer to the natural image.

**Table 1**. The comparison of PSNR(dB) and SSIM (add noise)

| | Set5 | | | Set14 | | |
|---|---|---|---|---|---|---|
| | ×2 | ×3 | ×4 | ×2 | ×3 | ×4 |
| **bicubic** | 33.05/0.899 | 29.65/0.812 | 28.32/0.754 | 29.75/0.822 | 26.90/0.709 | 25.49/0.623 |
| **ScSR[6]** | 34.89/0.921 | 31.21/0.848 | 28.61/0.765 | 30.79/0.849 | 27.64/0.729 | 25.64/0.635 |
| **ANR[7]** | 35.29/0.912 | 31.49/0.851 | 29.22/0.793 | 31.45/0.861 | 28.26/0.753 | 26.24/0.647 |
| **A+[11]** | 35.89/0.914 | 32.18/0.862 | 29.84/0.797 | 31.89/0.863 | 28.72/0.759 | 26.67/0.658 |
| **SISR[12]** | 34.69/0.923 | ~ | ~ | 30.89/0.859 | 28.17/0.746 | 26.08/0.651 |
| **SRCNN[8]** | 35.98/0.931 | 32.33/0.872 | 30.05/0.799 | 31.96/0.868 | 28.90/0.765 | 26.70/0.667 |
| **FSRCNN[18]** | 36.03/0.934 | 32.32/**0.875** | 30.07/0.803 | 31.98/0.867 | 28.91/0.764 | 26.69/0.665 |
| **our-s** | 36.33/0.941 | 32.45/0.873 | 30.26/0.811 | 32.23/0.877 | 29.07/0.771 | 26.95/0.675 |
| **our** | **36.41/0.942** | **32.47**/0.873 | **30.33/0.815** | **32.29/0.877** | **29.15/0.773** | **27.06/0.677** |

**Table 2**. Compare costs and runtimes with state-of-the-art deep learning and non-deep learning methods (without noise)

| | Deep learning SR methods | | | | Non-deep learning SR methods | | | our |
|---|---|---|---|---|---|---|---|---|
| | **SRCNN[8]** *ECCV14* | **FSRCNN[18]** *CVPR16* | **VDSR[19]** *CVPR16* | **lapSCN[20]** *CVPR17* | **MMPM[22]** *TIP18* | **WSD-SR[23]** *TIP18* | **SWL-SR[24]** *TIP18* | |
| **Costs**(The operations is computed for the situation that ×4 upsampling for 720p HR images) | | | | | | | | |
| **Parameters** | 60K | 40K | 665K | 813K | ~ | ~ | ~ | 5K |
| **Operations** | 75G | 8G | 612.6G | 29.9G | ~ | ~ | ~ | 0.6G |
| **Average runtime**(×4 upsampling for 256×256 HR images) | | | | | | | | |
| **CPU** | 1.8s | 0.6s | 7.9s | 1.7s | 5.5s | 9.8s | 96s | 0.08s |
| **GPU** | 0.13s | 0.06s | 0.1s | 0.09s | ~ | ~ | ~ | 0.003s |
| **Average PSNR**(dB) | | | | | | | | |
| **Set14 ×2** | 32.42 | 32.63 | 33.05 | 33.08 | 32.82 | 32.83 | 32.56 | 32.69 |
| **B100 ×4** | 26.91 | 26.95 | 27.29 | 27.32 | 26.85 | ~ | 27.06 | 26.98 |

In terms of cost and runtime, compared to deep learning SR methods that require multi-layer or even multi-channel networks, our method only uses several 1-or 2-hidden-layer BP networks to complete LR patch classification and upsampling, which can achieve 0.08s per 256×256 image for ×4 upsampling even on normal CPUs (matlab environment). Our method also guarantees the performance attribute to the proper classification scheme and the strong generalization ability of m-NNs. Table 2 shows the comparison with the state-of-the-art methods [8, 18, 19, 20, 21], we quantify the computation costs by the number of parameters and operations, and record the runtimes on GPUs and CPUs. It shows a significant advantage for our method on costs and speed. Besides, our average PSNR outperform other non-deep learning methods [22, 23, 24] and is closer to deep learning ones.

## 5. CONCLUSION

In this paper, we propose a real-time and efficient SR method which combines a sample classification tree based on SIMK and a mapping-learning stage based on nonlinear regression, which simultaneously guarantees both reconstruction quality and speed at the stage of classification and mapping-learning. Comparing with state-of-the-art methods, our method achieves both visual and performance improvement.

## 6. REFERENCES


[1] 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, 2010.

[2] H.S.Hou and H.C.Andrews, "Cubic splines for image interpolation and digital ltering," *IEEE Trans. Image Processing*, vol. 26, no. 6, pp. 508–517, 1978.

[3] Xuesong Zhang, Jing Jiang, Junhong Li, and Silong Peng, "Manifold learning-based sample selection method for facial image super-resolution," *Optical Engineering*, vol. 51, no. 4, 2012.

[4] Xuesong Zhang, Jing Jiang, and Silong Peng, "Commutability of blur and afne warping in super-resolution with application to joint estimation of triple-coupled variables," *IEEE Trans. Image Processing*, vol. 21, no. 4, pp. 1796–1808, 2012.

[5] H.Takeda, S.Farsiu, and P.Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Trans. Image Processing*, vol. 16, no. 2, pp. 349–366, 2007.

[6] J.Yang, J.Wright, T.S.Huang, and Y.Ma, "Image super-resolution via sparse representation," *IEEE Trans. Image Processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[7] R.Timofte, V.De, and L.Van Gool, "Anchored neighborhood regression for fast example-based super-resolution," *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 1920–1927, 2013.

[8] C.Dong, C.C.Loy, K.He, and X.Tang, "Learning a deep convolutional network for image super-resolution," *Proc. 13th Eur. Conf. Comput. Vis.*, pp. 184–199, 2014.

[9] H.Chang and D.Y.Yeung, "Super-resolution through neighbor embedding," 2004.

[10] B.Li, H.Chang, S.Shan, and X.Chen, "Locality reserving constraints for super-resolution with neighbor embedding," *Proc. 16th IEEE Int. Conf. Image Process.*, pp. 1189–1192, 2009.

[11] R.Timofte, V.De.Smet, and L.Van.Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," *Proc. Asian Conf. Comput. Vis.*, pp. 111–126, 2014.

[12] J.S.Choi and M.Kim, "Super-interpolation with edge-orientation-based mapping kernels for low complex 2 upscaling," *IEEE Trans. Image Processing*, vol. 25, no. 1, pp. 469–483, 2016.

[13] Y.Romano, J.Isidoro, and P.Milanfar, "Raisr: Rapid and accurate image super resolution," *IEEE Trans. Com utational Imaging*, vol. 3, no. 1, pp. 110–125, 2017.

[14] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita, "Deep back-projection networks for super-resolution," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018.

[15] Peiqi.Duan, Anlong.Ming, Xuejing.Kang, and Chao.Yao, "A new single image super-resolution method using simk-based classification and isrm technique," *International Conference on Pattern Recognition*, 2018.

[16] P.Arbelaez, M.Maire, C.Fowlkes, and J.Malik, "Contour detection and hierarchical image segmentation," *IEEE TPAMI*, vol. 33, no. 5, 2011.

[17] J.Pan, Z.Hu, Z.Su, and M.-H.Yang, "$l_0$-regularized intensity and gradient prior for deblurring text images and beyond," 2017, vol. 39, pp. 342–355.

[18] Chao Dong, Chen Change Loy, and Xiaoou Tang, "Accelerating the super-resolution convolutional neural network," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016.

[19] J.Kim, J.Kwon Lee, and K.Mu Lee, "Accurate image super-resolution using very deep convolutional networks," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016.

[20] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017.

[21] Z.Wang, D.Liu, J.Yang, W.Han, and T.Huang, "Deep networks for image super-resolution with sparse prior," *IEEEInternational Conference on Computer Vision*, 2015.

[22] Y. Huang, J. Li, X. Gao, L. He, and W. Lu, "Single image super-resolution via multiple mixture prior models," *IEEE Transactions on Image Processing*, vol. 27, no. 12, pp. 5904–5917, 2018.

[23] C. Cruz, R. Mehta, V. Katkovnik, and K. O. Egiazarian, "Single image super-resolution based on wiener filter in similarity domain," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1376–1389, 2018.

[24] Y. Li, W. Dong, X. Xie, G. Shi, J. Wu, and X. Li, "Image super-resolution with parametric sparse model learning," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4638–4650, 2018.