**A Heterogeneous Group CNN for Image Super-Resolution**

Chunwei Tian®, Member, IEEE, Yanning Zhang®, Senior Member, IEEE, Wangmeng Zuo®, Senior Member, IEEE, Chia-Wen Lin®, Fellow, IEEE, David Zhang®, Life Fellow, IEEE, and Yixuan Yuan®, Member, IEEE

Abstract—Convolutional neural networks (CNNs) have obtained remarkable performance via deep architectures. However, these CNNs often achieve poor robustness for image super-resolution (SR) under complex scenes. In this article, we present a heterogeneous group SR CNN (HGSRCNN) via leveraging structure information of different types to obtain a high-quality image. Specifically, each heterogeneous group block (HGB) of HGSRCNN uses a heterogeneous architecture containing a symmetric group convolutional block and a complementary convolutional block in a parallel way to enhance the internal and external relations of different channels for facilitating richer low-frequency structure information of different types. To prevent the appearance of obtained redundant features, a refinement block (RB) with signal enhancements in a serial way is designed to filter useless information. To prevent the loss of original information, a multilevel enhancement mechanism guides a CNN to achieve a symmetric architecture for promoting expressive ability of HGSRCNN. Besides, a parallel upsampling mechanism is developed to train a blind SR model. Extensive experiments illustrate that the proposed HGSRCNN has obtained excellent SR performance in terms of both quantitative and qualitative analysis. Codes can be accessed at https://github.com/hellloxiaotian/HGSRCNN.

Index Terms—Heterogeneous group convolutional architecture, image super-resolution (SR), multilevel enhancement mechanism, symmetric architecture.

I. INTRODUCTION

**S**INGLE image super-resolution (SISR) aims to obtain more natural and realistic textures from a given low-resolution (LR) image to its high-resolution (HR) image, which is very beneficial to high-level tasks, i.e., image classification [1] and object detection [2]. Due to ill-posed inverse characteristic, the SISR techniques have obtained enormous success via a degradation model with a priori knowledge, i.e., \( L = H_s \), where \( L \) and \( s \) represent an LR image and a scale factor, respectively [3]. Also, \( H \) denotes a predicted high-definition image. According to that, the SISR methods can be summarized into three paradigms in general, i.e., interpolation methods, optimization methods, and discriminative learning methods. Interpolation methods mainly relied on bilinear [4] or bicubic interpolation operations [5] to obtain a mapping from an LR image to an HR image. Although these methods were simple and efficient, they have obtained poor performance in SISR. To address this issue, optimization methods can be used to guide an SR model via natural image characteristics in a priori knowledge manner [6]. For instance, using a sparse priori knowledge to obtain a linear combination can effectively predict HR images [7]. However, this optimization method may enjoy a flexible work mode at the cost of a time-consuming process. Also, these methods may refer to manual setting parameters to achieve competitive SR performance. As an alternative, since discriminative learning methods have efficiency and flexibility, they are developed [8]. Notably, due to flexible end-to-end architectures, convolutional neural networks (CNNs) have dramatic demands in SISR [9]. The mentioned research can be generalized on two aspects in general, containing SR methods-based high-frequency and low-frequency information. The SR methods-based high-frequency information requires size consistency of input and output in a CNN, which results in given LR images need be converted to high-frequency images through a bicubic operation as training images for constructing an SR model [10]. Inspired by that,
a very deep SR network architecture was implemented using residual learning operations and stacking small filter sizes to obtain good visual effects [11]. Due to deep architectures, CNNs are faced with training difficulty. To overcome the mentioned problem, recursive learning and residual learning techniques are presented to accelerate training speed [12], [13]. For instance, a deeply recursive convolutional network (DRCN) integrated hierarchical information via residual learning techniques to facilitate accurate features for preventing exploding and vanishing gradients [12]. Besides, fusing global and local information through skip connections to guide a new network architecture can enhance the learning ability for SISR [14]. As an alternative, exploiting new components (i.e., recursive unit and gate unit) to obtain multilevel representation can improve the quality of a predicted image [15]. Although these approaches can outperform traditional methods in SISR, they may refer to high complexity [16]. To overcome the challenge, the SR methods-based low-frequency information is developed. That is, directly inputting LR image into a CNN and using an upsampling operation of deep layer to amplify obtained low-frequency features can train an SR model [17]. For example, designing a deformable and attentive mechanism to enhance a CNN extracted salient low-frequency texture information to enhance visual effects [17]. Although the mentioned methods have achieved remarkable SR results, they only roughly fuse hierarchical features via residual learning or concatenation operations to affect different layers. These results in obtained features of simplification cannot represent well high-quality images, which achieves poor robustness in SISR under complex scenes.

In this article, we propose a heterogeneous group SR CNN (HGSRCNN). It mainly uses heterogeneous group blocks (HGBs) to integrate structure information of different types for obtaining an HR image. Each HGB uses a heterogeneous architecture composing of a symmetric group convolutional block and a complementary convolutional block via enhancing internal and external relations of different channels in a parallel way to obtain more representative structure information of different types. Also, a refinement block (RB) with signal enhancement ideas in a serial way is developed to remove useless information for accelerating training efficiency. To alleviate loss of original information problem, a multilevel enhancement mechanism guides a CNN to construct a symmetric architecture for progressively facilitating the information of HGSRCNN in SISR. In addition, a parallel upsampling mechanism is developed to train a blind SR model.

Main contributions of proposed HGSRCNN are conducted as follows.

1) The proposed 52-layer HGSRCNN uses heterogeneous architectures and RBs to enhance internal and external interactions of different channels both in parallel and serial ways for obtaining richer low-frequency structure information of different types, which is very suitable to SISR in complex scenes.

2) A multilevel enhancement mechanism guides a CNN to implement a symmetric architecture for progressively facilitating structural information in SISR.

3) The designed HGSRCNN obtains competitive execution speed for SISR. That is, it only takes the run time to 4.58% of RDN and 4.27% of SRFBN in restoring a high-quality image with $1024 \times 1024$.

The remainder of this article is conducted as follows. Section II reveals related work of the proposed method. Section III illustrates our proposed method. Section IV gives experimental analysis and results. Section V concludes the proposed method.

II. RELATED WORK

A. Enhancement of Different Structure Features for SISR

Due to strong expressive ability, CNNs become popular in SISR. Notably, remarkable performance of CNNs is affected by deeper network architectures. To address this issue, enhancing structure features of deep networks can improve the interaction of both shallow and deep layers. Mentioned techniques are usually classified into two categories: enhancements of high-frequency and low-frequency structure features.

Enhancements of high-frequency structure features are composed of two stages [18]. The first stage utilizes bicubic-interpolation or bilinear-interpolation operations to zoom corrupted low-resolution images as high-frequency images. Then, using residual learning or concatenation operations integrates these high-frequency features via a designed CNN to facilitate richer structure features. Inspired by that, Kim et al. [11] proposed a deeper network based on VGG via some small filters to obtain high-frequency structure information, and a residual learning operation is used to enhance obtained structure information in SISR. Subsequently, a recursive CNN transferred structure features of shallow layers to the final layer through skip connections in a shared-parameter manner for enhancing the clarity of predicted images [12]. Alternatively, a multipath residual CNN used global and local residual learning operations to fuse hierarchical structure features for improving a learning ability of a deep network in SISR [13]. Besides, using skip-layer connections to connect multiple convolutional and deconvolutional layers for implementing a symmetrical network can also obtain more detailed structure features in SISR [14]. Although these mentioned techniques have achieved excellent SR performance, they are faced with huge computational costs caused by training images of large sizes. To resolve this problem, enhancement methods of low-frequency structure features are presented.

Enhancements of low-frequency structure features directly input corrupted low-resolution images into a CNN via a residual learning operation to extra robust low-frequency structure features; then, they can use an upsampling technique to deal with obtained low-frequency structure features for predicting high-quality images [19]. For instance, a residual dense network repeatedly used residual learning techniques to enhance effects of each layer for extracting more accurate low-frequency structure features in SISR [20]. Alternatively, a multipath residual network can aggregate different hierarchical features via different paths to enhance the robustness of obtained low-frequency structure information in SISR [21]. Besides, to accelerate the training efficiency, more refinement
networks are presented [22]. A cascade network with many smaller filters (i.e., convolutions of $1 \times 1$) utilized multiple shortcut connections to efficiently mine structure information of different types for obtaining a strong expressive ability of an SR model [22]. Along this line, a coarse-to-fine SR CNN (CFSRCNN) applied residual learning and concatenation techniques in a heterogeneous architecture to, respectively, enhance low- and high-frequency structure information to improve the training stability and pursue excellent SR performance [23]. The mentioned research shows that integrating different structure features is beneficial to SISR. Motivated by that, we design a diversified network architecture to enhance the effect of both internal connections from structure information of the same level and contextual structure information for SISR in this article, according to training strategies of deep networks and signal processing knowledge.

B. Deep CNNs-Based Different Channels for SISR

Due to training difficulty of deeper network architectures, applying residual operation and skip connection techniques to transmit memory of shallow layers for obtaining more details of high-quality images are proposed [24]. It is noted that these methods roughly fused hierarchical features to promote generalization abilities of SR models. They may cause a large computational burden. To address the phenomenon, deep CNNs with different channels are explored for SISR [25]. Referred methods can be roughly summarized into two categories: local channels and global channels.

The first strategy mentioned earlier splits all the channels via attention techniques to extract salient information for highlighting key channels in SISR, which can improve the training efficiency of SR models [26]. For instance, Zhang et al. [26] employed a residual channel attention mechanism to reinforce the interdependencies of different channels and adaptively filtered abundant low-frequency features for improving the performance of SISR. Besides, Niu et al. [27] applied a holistic attention mechanism consisting of a layer attention module and a channel-spatial attention module to strengthen the correlations among different layers, channels, and positions for selecting more expressive information in SISR.

The second mentioned strategy directly merged hierarchical channel information via residual learning and skip connection techniques to mine rich low-frequency features in SISR [28]. The simultaneous use of a dense network structure with group convolutions and small filters of $1 \times 1$ can enhance the relationships of different channels via removing redundant parameters to progressively obtain useful information in SISR [28]. Along this line, Jain et al. [29] unified group convolutional techniques and pruning ideas into a frame via throwing away useless information to reduce the test time of predicted HR images. Alternatively, aggregating obtained features from splitting convolutions by all the steps efficiently extracted discriminative information, i.e., edges, corners, and textures to achieve clearer visual effects [30]. Besides, dividing a CNN into two subnetworks via a splitting operation to, respectively, learn robust hierarchical channel features can facilitate complementarity of different channels in MR image super-resolution (SR) [31].

The research mentioned earlier illustrated that aggregating different information to enhance the interaction among different channels can achieve excellent SISR performance. Motivated by that, we design a symmetric architecture via two twin branches to strengthen inner connections of different channels. Besides, we offer a supplementary block to learn features of all the channels, which can implement a complementary of both internal and external channels to obtain richer structure information in SISR. More information is given in Section III.

III. PROPOSED METHOD

The proposed HGSRCNN includes four components: two convolutional layers with rectified linear unit (ReLU) [32], six symmetrical HGBs as well as HGBs, a parallel upsampling mechanism, and a single convolutional layer, as illustrated in Fig. 1. Specifically, each symmetrical HGB uses a symmetric group convolutional block and a complementary convolutional block in a parallel way to enhance internal and external relations of different channels for facilitating richer low-frequency structure information. Taking redundant features of the mentioned enhancement operation and training of deep CNNs into account, an RB with signal enhancement ideas is used into an HGB to remove useless information for accelerating the training. To prevent loss of original information, two enhancement branches are embedded into these HGBs to implement a local symmetrical architecture for progressively gathering low-frequency features in SISR. Besides, a parallel upsampling with multiple scales is used to train a blind SR model. Finally, a signal convolutional layer is employed to construct an HR image. More contents of the proposed HGSRCNN are given in the latter parts. The proposed 52-layer incorporates two-layer convolutions with ReLUs, 48-layer HGB, one-layer parallel upsampling mechanism, and a single convolutional layer. The mentioned two-layer convolutions with ReLUs are set as the 1st and 50th layers. Each convolution with an ReLU is equal to a convolutional layer that acts a ReLU, which can be regarded as Conv $+$ ReLU in Fig. 1. It can be observed that the first Conv $+$ ReLU can obtain low-frequency features from an observation LR image through a convolutional operation; then, obtained linear features can be mapped into a nonlinearity via an activation function of an ReLU. Also, its parameters are set to the input channels of 3, a filter size of $3 \times 3$, and the output channels of 64. Subsequently, six HGBs can extract richer low-frequency context structure information via enhancing internal and external relations of different channels in a parallel and serial way to obtain excellent SR performance. Parameters of each HGB are fixed as the input channels of 64, a filter size of $3 \times 3$, and the output channels of 64, respectively. To prevent loss of original information, two enhancement branches are embedded into these HGBs to implement a local symmetrical architecture for progressively gathering low-frequency features in SISR, as described in Fig. 1. To avoid over-enhanced phenomenon from two enhancement branches mentioned earlier, the second Conv $+$ ReLU is used to remove redundant low-frequency
features, where its parameters are the same as HGB. That is, two enhancement branches (multilevel enhancement mechanism) act the ends of both the first layer and the sixth HGB and the ends of the second and fifth HGBs through residual learning operations, respectively. Besides, a parallel upsampling mechanism can be exploited to map obtained low-frequency features into high-frequency features. It is noted that the referred to technique can simultaneously execute three different scales (i.e., $\times 2$, $\times 3$, and $\times 4$) via a switch to train a blind model. Also, they enjoy the same setting as each HGB. Finally, a signal convolutional layer is conducted to obtain a predicted high-quality image through obtained high-frequency features. Its input and output channel number of 3 and a filter size of $3 \times 3$ are given as the parameters of the final layer. To conveniently understand the working procedure of HGSRCNN, some characters are given. Let $P_{\text{LR}}$ and $O_1 = R(C(P_{\text{LR}}))$. Besides, HGSRCNN is the function of HGSRCNN, which can be optimized via the following objective function.

A. Loss Function

To fairly optimize the parameters of HGSRCNN, a mean squared error (MSE) [18], [33] is selected as a loss function to train an HGSRCNN model in SISR. The HGSRCNN first uses a given LR image $P_{\text{LR}}$ as the input of HGSRCNN to obtain the predicted HR image $P_{\text{SR}}$. Then, using MSE to compute the difference between the obtained HR image $P_{\text{SR}}$ and a given HR image $P_{\text{HR}}$ can optimize parameters. This process can be formulated as follows:

$$L_0(p) = \frac{1}{2N} \sum_{j=1}^{N} \| \text{HGSRCNN}(P_{\text{LR}}^j) - P_{\text{HR}}^j \|^2$$  \hspace{1cm} (2)

where $L_0$ is the loss function of MSE, and $P_{\text{LR}}^j$ and $P_{\text{HR}}^j$ are the $j$th LR and HR training images, respectively. Besides, $N$ denotes the number of training images. $p$ is treated as the parameter set of training an HGSRCNN model.

B. Heterogeneous Group Block

An eight-layer HGB is used to facilitate more representative structure information of different types via a novel heterogeneous architecture to enhance the internal and external relations of different channels for improving SR performance.
Heterogeneous Group Block: The three-layer heterogeneous convolutional block is composed of a symmetric group convolutional block, and a complementary convolutional block is used to enhance the internal and external relations of different channels for extracting robust low-frequency structure information. In terms of internal relation enhancement of different channels, two three-layer subnetworks in the symmetric group convolutional block, respectively, learn representative information of split channels and integrate obtained features via a concatenation operation to enhance their internal correlations in SISR. Specifically, each layer of each subnetwork is Conv + ReLU. Also, the input and output channels of each layer are 32, respectively. Their filter sizes are $3 \times 3$. Besides, output channel of symmetric group is 64, which is obtained by concatenating outputs of two subnetworks. To visually explain the mentioned process, the following formulas can be given.

First, we use a splitting operation to divide input of current heterogeneous group block into two parts ($I^U_i$ and $I^L_i$) as inputs of two subnetworks in the symmetric group convolutional block, as shown in (3) and (4), where $I^U_i$ and $I^L_i$ are the upper half and the lower half of all the channel features, respectively.

\[
I^U_i = \begin{cases} 
U, & O_1 \\
\frac{U}{2}, & O_{i-1}
\end{cases}
\]

\[
I^L_i = \begin{cases} 
L, & O_1 \\
\frac{L}{2}, & O_{i-1}
\end{cases}
\]  

(3)  

(4)

where $O_{i-1}$ denotes the output of the $i - 1$th layer, and $i \geq 2$. Specifically, $O_1$ expresses the output of the first layer in the HGSRCNN. Also, $O_{i-1}$ denotes the output of the $i - 2$th HGB. Also, $(U/2)$ and $(P/2)$ are defined as a splitting operation from channels of the upper half and the lower half, respectively. The obtained $I^U_i$ and $I^L_i$ act two subnetworks of a symmetric group convolutional block, as illustrated in the following:

\[
O_i^{SGCB} = R(Concat(C(R(C(R(C(I^U_i))))), C(R(C(R(C(I^L_i)))))))
\]  

(5)

where $O_i^{SGCB}$ is the output of symmetric group convolutional block in the $i - 1$th HGB ($2 = < i <= 7$), and Concat denotes a concatenation operation as presented in Fig. 1. Also, the output channel number of $O_i^{SGCB}$ is 64.

Taking entirety of all channels into consideration, a three-layer complementary convolutional block is designed to enhance their external correlations for enhancing the robustness of obtained features in SISR, which is complementary to the symmetric group convolutional block. Each layer of complementary convolutional block is composed of Conv + ReLU. Also, the parameters of each layer are an input channel of 64, an output channel of 64, and a filter size of $3 \times 3$. Besides, the output of a heterogeneous convolutional block can be obtained by a residual learning to fuse outputs of both symmetric group convolutional block and complementary convolutional block as an input of an RB. The procedure can be formulated as follows:

\[
O_i^{CCB} = R(C(R(C(R(C(I_i))))))
\]  

(6)

where $I_i$ is the output of the upper half of the RB. When the upper layer is the first layer, $I_1 = O_1$. Otherwise, $I_i$ is $O_{i-1}$ ($2 = < i <= 7$). Also, $O_i^{CCB}$ is the output of a complementary convolutional block from the $i - 1$th HGB. Subsequently, a residual learning operation is used to fuse the outputs of a symmetric group convolutional block and a complementary convolutional block as an output of heterogeneous convolutional block as follows:

\[
O_i^{HCB} = O_i^{SGCB} + O_i^{CCB}
\]  

(7)

where $O_i^{HCB}$ expresses the output of heterogeneous convolutional block in the $i - 1$th HGB, which acts an RB. Also, + denotes a residual learning operation, which is equal to $\oplus$ in Fig. 1.

Refinement Block: To reduce the importance of redundant information from heterogeneous convolutional block, a five-layer RB is designed. Each layer of the RB is composed of Conv + ReLU, and their parameters are the input channel of 64, the output channel of 64, and a filter size of $3 \times 3$. To strength the memory ability of shallow layers on deep layers in SISR, we use signal enhancement operations into the RB. That is, signal enhancement operations include a global signal enhancement and a local signal enhancement. The global signal enhancement utilizes a residual learning technique to enhance the entire structure information from heterogeneous convolutional block, a five-layer convolutional block as an output of heterogeneous convolutional block, a five-layer RB is designed. Each layer of the RB is composed of Conv + ReLU, and their parameters are the input channel of 64, the output channel of 64, and a filter size of $3 \times 3$. To strength the memory ability of shallow layers on deep layers in SISR, we use signal enhancement operations into the RB. That is, signal enhancement operations include a global signal enhancement and a local signal enhancement. The global signal enhancement utilizes a residual learning technique to fuse the output of the heterogeneous convolutional block and output of the RB. The local signal enhancement utilizes a residual learning technique to integrate the output of the first layer in the RB and output of the RB. The implementation can be expressed as follows:

\[
O_i^{HGB} = R(C(R(C(R(C(R(C(I_i)))))))) + R(C(O_i^{HCB}))
\]

\[
= \text{HGB}(O_i^{HCB})
\]  

(8)

where $O_i^{HGB}$ is the output of the $i - 1$th HGB.
C. Multilevel Enhancement Mechanism

To prevent the loss of original information, a multilevel enhancement mechanism is embedded into these HGBs via two enhancement branches to implement a local symmetrical architecture for progressively gathering low-frequency features in SISR, as described in Fig. 1. The first enhancement branch (global symmetrical enhancement) is that fuses the outputs of the first HGB and the fifth HGB via a residual learning operation as the input of the sixth HGB. The mentioned implementations can be described as follows:

\[
I_6 = GE_1(HGB_5(O_1)) = O_6^{HGB} + O_2^{HGB}
\]

(9)

where GE1 expresses a function of the first enhancement branch, and HGB5 is symboola as the functions of five HGBs. \(I_6\) is the input of the sixth HGB. Besides, \(O_6^{HGB}\) and \(O_2^{HGB}\) denote the outputs of the second and fifth HGBs, respectively. To further improve the importance of hierarchical features, the second enhancement branch (local symmetrical enhancement) is designed by a residual learning operation. The second enhancement branch acts both the first layer of HGSRCNN and the sixth HGB as follows:

\[
O_{HGBS} = GE_2(O_6^{HGB}) = O_1 + O_6^{HGB}
\]

(10)

where \(O_{HGBS}\) is the output of all the HGBs as the input of the second Conv + ReLU. GE2 denotes the function of the second enhancement branch. The second Conv + ReLU is used to prevent the over-enhancement phenomenon of HGBs, and it acts a parallel upsampling mechanism.

D. Parallel Upsampling Mechanism

Due to ill-posed inverse characteristic of image SR, scholars tend to establish an SR model via a certain scale. However, the LR images have suffered from different corruption, which makes most of the existing SR models cannot exert effects [34]. To resolve this issue, a parallel upsampling mechanism [22] with a flexible controller is used in the HGSRCNN to achieve a blind SR model. Its implementations and work mechanism can be illustrated as follows.

The parallel upsampling mechanism contains three components, i.e., \(\times 2\) Upsampling, \(\times 3\) Upsampling, and \(\times 4\) Upsampling. Specifically, \(\times 2\) Upsampling, \(\times 3\) Upsampling, and \(\times 4\) Upsampling can be, respectively, equal to a Conv + Shuffle \(\times 2\), Conv + Shuffle \(\times 3\), and Conv + Shuffle \(\times 4\) (also regarded as two Conv + Shuffle \(\times 2\)), where \(\times 2\) Upsampling and \(\times 3\) Upsampling denote a convolution with a size of \(3 \times 3\) and Shuffle \(\times 2\) and Shuffle \(\times 3\), respectively. Also, input and output channels of each component are 64. Besides, a flexible controller can control different components to obtain a blind SR model. That is, if the controller value is 0, three components will parallel work to train an SR model for different scales (i.e., \(\times 2\), \(\times 3\), and \(\times 4\)), as presented in Fig. 2, which is expressed by mentioned solid line part. Otherwise, the controller value can be extended to be a scale from 2, 3, and 4, and an SR model with a certain scale is obtained, which is represented by mentioned dotted line part in Fig. 2. To intuitively show the execution process, the following equation is conducted:

\[
O_{PUM} = PUM(O_{HGBS}) = \begin{cases} 
PS_2(C(O_2^{HGBS})) \circ PS_3(C(O_3^{HGBS})) & \text{if } i = 0 \\
PS_2(C(O_2^{HGBS})) \circ PS_3(C(O_3^{HGBS})), & \text{if } i = 2, 3, 4 
\end{cases}
\]

(11)

where \(O_{PUM}\) denotes the output of the parallel upsampling mechanism. \(O_2^{HGBS}\), \(O_3^{HGBS}\), and \(O_4^{HGBS}\) are used to stand for outputs of obtained low-frequency structure information for \(\times 2\), \(\times 3\), and \(\times 4\), respectively. Also, \(PS_2\), \(PS_3\), and \(PS_4\) are symbolized as the functions of \(\times 2\) Upsampling, \(\times 3\) Upsampling, and \(\times 4\) Upsampling, respectively. Let \(\circ \) express a parallel operation. \(O_2^{HGBS}\) and \(PS_i\) are used to represent low-frequency output and function of an upsampling operation for a scale factor with \(i\), respectively. Besides, \(O_{PUM}\) acts a single convolutional layer as the last layer in the HGSRCNN, as given in (12), which can be utilized to construct predicted high-quality images. Its parameters are the input channel number of 64, the output channel number of 3, and a filter size of \(3 \times 3\)

\[
P_{SR} = C(O_{PUM}).
\]

(12)

IV. EXPERIMENTS

A. Training Datasets

To guarantee experimental fairness, a popular color image dataset of DIV2K [34] is used to train an HGSRCNN model. The DIV2K contains the training samples of 800 natural images, the validation samples of 100 natural images, and the test samples of 100 natural images for different scales in \(\times 2\), \(\times 3\), and \(\times 4\). Besides, to make obtained SR model more robust, the following data augment way is exploited to enlarge the training dataset [23]. First, the training dataset and the validation dataset from the same scale are merged into a new training dataset for training an HGSRCNN model. Secondly,
to improve the training efficiency of an HGSRCNN model, each LR image is cropped as the patches of size $81 \times 81$. Finally, random horizontal flips and rotation operation of $90^\circ$ are used to deal with these patches for extending categories of training samples.

B. Testing Datasets

Inspired by popular SR methods (i.e., LESRCNN [18], CARN [22], and CFSRCNN [23]), four public datasets containing Set5 [35], Set14 [35], BSD100 (B100) [36], and Urban100 (U100) [37] of $\times 2$, $\times 3$, and $\times 4$ are used as test datasets. The Set5 and Set14 have, respectively, captured 5 and 14 color images via the same digital devices for three scales ($\times 2$, $\times 3$, and $\times 4$). B100 and U100 include the 100 color images for $\times 2$, $\times 3$, and $\times 4$, respectively.

Motivated by the state-of-the-art SR methods [18], [22], Y channel in YCbCr space is chosen to conduct experiments in this article. That is, the predicted RGB images of an HGSRCNN model need be converted as the images of Y channel to test the performance of a designed HGSRCNN for image SR.

C. Experimental Settings

To better train a blind model, initial parameters are given as follows. Initial learning rate is set to $1e-4$, which may be halved for every $4e + 5$ steps from 553,000 steps. Also, a batch size is treated as 32, an epsilon of $1e - 8$, a $\beta_1$ of 0.9, a $\beta_2$ of 0.999, and more initial parameters are referred to [18] and [22]. Also, the controller value of 0 has conducted experiments in this article. Besides, training parameters are updated via an optimizer of Adam [38].

The HGSRCNN network is implemented by Pytorch of 1.2.0, Python of 3.6.6 on the Ubuntu system of 16.04. Besides, a PC containing RAM of 16 GB, one graphic processing unit (GPU) [39] with Inter Core i7-7800, and two GPUs with Nvidia GeForce GTX 1080Ti is used to provide computational ability. Specifically, mentioned GPUs reply on Nvidia CUDA of 11.3 and cuDNN of 8.0 to improve the execution speed.

D. Network Analysis

As is known to all, merging hierarchical features can enhance the importance of shallow layers on deep layers to promote the SR performance [15]. However, most of these SR techniques roughly merge obtained features of all the channels rather than strengthening the effects of local salient channels, which may result in obtained information of simplification cannot completely express high-quality images and achieve poor robustness for SISR of complex scenes. To address this issue, we present an HGSRCNN via integrating structure information of different types to enhance relations of different channels. Specifically, an HGB uses a symmetric group convolutional block and a complementary convolutional block to enhance internal and external relations of different channels to obtain more expressive structure information of different types. Also, an RB with signal enhancement ideas is fused into an HGB to filter useless information for accelerating training efficiency. To alleviate original information loss problem, a multilevel enhancement mechanism guides a CNN to construct a symmetric architecture via different HGBs and residual learning operations for progressively facilitating information of HGSRCNN in SISR. In addition, a parallel upsampling mechanism is developed to train a blind SR model. More detailed information of HGSRCNN in design principle of a network architecture and effectiveness of important components are described as follows.

HGSRCNN contains two Conv + ReLU, several HGBs, and a parallel upsampling mechanism, and a single Conv. The first Conv + ReLU can be employed to convert a given low-resolution image into nonlinear low-frequency features. According to the VGG architecture [40], increasing the depth of network can mine more useful features. Inspired by that, six HGBs are stacked behind the first Conv + ReLU to extract richer low-frequency structure information. The design rules of each HGB are conducted via enhancing relations of different channels and training strategies of deep networks.

Relations of Enhancing Different Channels: To enhance the expressive ability, some methods only roughly fuse hierarchical features through a residual learning or concatenation operations to strengthen the effects of different layers. However, due to simplification of obtained features, they cannot represent well high-quality images, which obtained poor robustness in SISR under complex scenes. To address this problem, we design a heterogeneous architecture via enhancing internal and external relations of different channels to extract more accurate low-frequency features.

In terms of enhancing internal relations of different channels, we propose a three-layer symmetric group convolutional block. The mentioned symmetric group convolutional block first halves output channels of the last HGB into two parts as inputs of two subnetworks. Next, each subnetwork is used to learn more accurate low-frequency channel structure information, respectively. Finally, a concatenation operation is used to merge obtained features from two subnetworks. Although the mentioned mechanism can enhance the internal relations of different channels, they ignore overall of obtained features from all the channels. Increasing the width of a deep network can capture more complementary information, according to the GoogLeNet [41]. Inspired by that, we design a three-layer complementary convolutional block to strengthen external relations of different channels for mining complementary low-frequency structure features, which makes an obtained SR model robust for complex scenes. Because the complementary convolutional block and the symmetric group convolutional block are parallel executed, the process is treated as a parallel procedure. Besides, we conduct a Table I to verify the effectiveness of the mentioned blocks. That is, a seven-layer normal convolutional network (NCN) can be better than a symmetric group convolutional network (SGCN) in peak signal-to-noise ratio (PSNR) [42] and structural similarity index (SSIM) [42] on U100 for $\times 2$, where NCN denotes a combination of five-layer Conv + ReLU, one-layer parallel upsampling mechanism, and one-layer Conv, and the SGCN denotes a combination of three-layer symmetric group convolutional block, one-layer Conv + ReLU, one-layer parallel upsampling mechanism, and one-layer Conv.
Also, the number of parameters from the SGCN is 71% of the NCN, as shown in Table II. In a summary, the proposed symmetric group convolutional block can make a trade-off between performance and complexity. Besides, the HSRCNN without GSE, LSE, LOSE, and RB has remarkable improvement than that of HGSRCNN without GSE, LSE, LOSE, and RB, and complementary convolutional block (CCB) in both PSNR and SSIM on U100 for ×2, as shown in Table I, where GSE and LSE denote a global symmetrical enhancement and a local symmetrical enhancement, respectively. Also, the HGSRCNN without GSE, LSE, LOSE, and RB is more superior to that of NCN on U100 for ×2, as illustrated in Table I, where RB expresses an RB, and LOSE denotes a local signal enhancement. That shows that the combination of the proposed symmetric group convolutional block and complementary convolutional block is more effective in SISR. Although the heterogeneous convolutional block can enhance the relations of different channels, they may include redundant information to affect the training speed.

Refinement Block: To resolve this problem, we propose an RB to learn more accurate low-frequency structure information in a serial way. Also, increasing the depth of a deep network can enlarge receptive field to mine more useful information, according to VGG [40]. Motived by that, a stacked five-layer Conv + ReLU forms a RB. Its effectiveness is proven by HGSRCNN without GSE, LSE, LOSE, and RB and HGSRCNN without GSE, LSE, and local signal enhancement (LOSE) in Table I. Besides, it is known that the depth of a deep network is bigger, and its performance may drop [15]. To tackle this problem, signal enhancement operations are gathered in an HGB.

The mentioned signal enhancement operations depend on two signal enhancements to strengthen the memory abilities of shallow layers on deep layers in SISR, according to training strategies of deep networks. That is, signal enhancement operations include a global signal enhancement (GOSE) and a local signal enhancement as well as LOSE. The GOSE utilizes a residual learning technique to fuse the input of the heterogeneous convolutional block and the output of the RB. The LOSE utilizes a residual learning technique to integrate output of the first layer in the RB and output of the RB. Also, their effective results are verified, as presented in Table I. That is, HGSRCNN without LSE and global symmetrical enhancement outperforms HGSRCNN without GSE, LSE, and local signal enhancement in both PSNR and SSIM on U100 for ×2. Although mentioned HGBs can effectively mine low-frequency information, they ignore relations of different HGBs. To prevent the phenomenon, a multilevel enhancement mechanism is designed as follows.

Multilevel Enhancement Mechanism: This mechanism relies on two enhancement branches to make HGSRCNN implement a local symmetrical architecture for progressively gathering low-frequency features for SISR, as described in Fig. 1. The first enhancement branch (local symmetrical enhancement) is that fuses the outputs of the first layer of the HGB and the fifth HGB via a residual learning operation as input of the sixth HGB. Its effect is tested via comparisons between HGSRCNN and HGSRCNN without local symmetrical enhancement in Table I. The second enhancement branch as well as global symmetrical enhancement is used to fuse outputs of the first layer of HGSRCNN and the sixth HGB. From Table I, we can see that the HGSRCNN with local symmetrical enhancement can obtain better results than that of HGSRCNN without LSE and global symmetrical enhancement, which also implies the importance of the GSE in SISR. In addition, to avoid the over-enhanced phenomenon from two enhancement branches mentioned earlier, a Conv + ReLU is used to remove redundant low-frequency features, where its parameters are the same as input channel number, output channel number, and filter size of each HGB. To handle blind SR, a parallel upsampling mechanism is used to implement an SR for multiple scales, as shown in (12). Besides, a convolutional layer is used as the last layer to obtain predicted high-quality images.

E. Comparisons With State-of-the-Arts

To evaluate the SR effects of HGSRCNN from different angles, this article conducts experiments in terms of quantitative and qualitative analyses. Specifically, the quantitative analysis is used to test SR results containing PSNR, SSIM, run time of restoring high-quality images, complexities, and perceptual quality of feature similarity index (FSIM) [63] of popular SR techniques, containing Bicubic [43], A+ [44], RFL [45], self-exemplars SR (SelfEx) [37], a denoising CNN (DnCNN) [47], the cascade of sparse coding-based networks (CSCN) [46], 30-layer residual encoder–decoder network (RED30) [14], trainable nonlinear reaction diffusion (TNRD) [48], fast dilated SR convolutional network (FDSR) [49], an SR CNN (SRCNN) [10], fast SR CNN

| Methods | PSNR (dB) | SSIM |
|---------|-----------|------|
| Normal convolutional network (NCN) | 30.59 | 0.9121 |
| Symmetric group convolutional network (SOGCN) | 30.42 | 0.9068 |
| HGSRCNN without GSE, LSE, LOSE, RB and complementary convolutional block (CCB) | 31.23 | 0.9156 |
| HGSRCNN without GSE, LSE, LOSE and enhancement branch (RB) | 31.74 | 0.9239 |
| HGSRCNN without GSE, LSE and local signal enhancement (LOSE) | 32.18 | 0.9253 |
| HGSRCNN without LSE and global symmetrical enhancement (GSE) | 32.17 | 0.9258 |
| HGSRCNN without local symmetrical enhancement (LOSE) | 32.20 | 0.9256 |
| HGSRCNN (Max) | 32.21 | 0.9259 |

| Methods | Parameters (M) | Flops (G) |
|---------|---------------|-----------|
| SGCN | 132.40K | 1.63G |
| NCN | 187.70K | 1.99G |

TABLE I

PSNR and SSIM of Different SR Methods on U100 for ×2

TABLE II

Complexity of Different SR Networks

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
(FSRCNN) [16], residue context network (RCN) [50], very deep SR network (VDSR) [11], deeply recursive convolutional network (DRCN) [12], context-wise network fusion (CNF) [51], Laplacian SR network (LapSRN) [52], information distillation network (IDN) [53], deep recursive residual network (DRRN) [13], balanced two-stage residual networks
TABLE VII
AVERAGE PSNR/SSIM VALUES OF DIFFERENT SR METHODS FOR ×4 ON B100

| Methods      | PSNR/SSIM          |
|--------------|--------------------|
| RDN [10]     | 21.72 / 0.9419     |
| CSRN [11]    | 22.76 / 0.9535     |
| SRFBN [19]   | 22.72 / 0.9409     |
| CFSRCNN [23] | 22.55 / 0.9335     |
| HGSRCNN (Ours) | 23.46 / 0.9563   |

TABLE VIII
RUNNING TIME (SECONDS) OF DIFFERENT SR METHODS ON PREDICTING HR IMAGES OF SIZES 256 × 256, 512 × 512, AND 1024 × 1024 FOR ×2

| Size          | Single Image Super-Resolution |
|---------------|-------------------------------|
|              | VDSR [11] | 0.0172 | 0.0575 | 0.2126 |
|              | DRRN [13] | 2.063  | 5.090  | 25.23  |
|              | MemNet [15] | 0.9794 | 3.263  | 14.69  |
|              | RDN [20]  | 0.0533 | 0.2232 | 0.9124 |
|              | SRFBN [19] | 0.0761 | 0.3568 | 0.9787 |
|              | CARN-M [22] | 0.0139 | 0.0919 | 0.0320 |
|              | CFSRCNN [23] | 0.0133 | 0.0184 | 0.0298 |
|              | ACNet [57] | 0.0166 | 0.0195 | 0.0315 |
|              | HGSRCNN (Ours) | 0.0234 | 0.0337 | 0.0418 |

TABLE IX
COMPLEXITIES OF DIFFERENT SR METHODS FOR ×2

| Methods      | Parameters | FLOPs | FLOPS |
|--------------|------------|-------|-------|
| VDSR [11]    | 355K       | 1.57M  | 15.80 |
| DRRN [13]    | 355K       | 1.57M  | 15.80 |
| MemNet [15]  | 355K       | 1.57M  | 15.80 |
| RDN [20]     | 355K       | 1.57M  | 15.80 |
| CARN-M [22]  | 841K       | 3.84M  | 38.79 |
| CFSRCNN [23] | 162K       | 1.01M  | 10.10 |
| ACNet [57]   | 74M        | 19.8M  | 198.6 |
| HGSRCNN (Ours) | 222K     | 24.5M  | 245.0 |

Fig. 3. Visual effect of different methods for ×3 on U100 as follows. (a) HR image, (b) VDSR, (c) DRCN, (d) CARN-M, (e) LESRCNN, (f) CFSRCNN, (g) ACNet, and (h) HGSRCNN (Ours).
superior to HGSRCNN in PSNR and SSIM on a dataset with large volume samples (B100) for ×4, as given in Table VII, they are faced to huger complexity and running time than that of HGSRCNN as reported in latter context. In a summary, these illustrations show that the proposed HGSRCNN obtains excellent performance to deal with LR images of different backgrounds.

It is known that digital devices have demands for execution time and complexity [23], [36]. According to that, we use eight popular SR methods, i.e., VDSR, DRRN, MemNet, RDN, SRFBN, CARN-M, CFSRCNN, and ACNet as comparative methods to restore high-quality images with 256 × 256, 512 × 512, and 1024 × 1024 on ×2 to test running time of these methods. As described in Table VIII, we can see that HGSR-CNN achieves execution fast in SISR. That is, HGSRCNN takes the run time to 4.58% of popular RDN and 4.27% of SRFBN in predicting an HR image with a size of 1024 × 1024. In terms of complexity, we exploit VDSR and DnCNN, DRCN, MemNet, CARN-M, CARN, CSFM, RDN, SRFBN, ACNet, and HGSRCNN to conduct experiments for measuring their complexities. Specifically, the numbers of parameters and flops [65] of training an SR model are used to express as complexity of computational cost and memory consumption for predicting SR images of size 162 × 162. As given in Table IX, HGSRCNN only takes the number of parameters to 9.9% of 134-layer RDN and 16.9% of 384-layer CSFM to obtain approximative SR results. Besides, Table IX reports that HGSRCNN only takes 10.40% of RDN and 17.70% of CSFM in flops. Thus, HGSRCNN is a useful SR tool in terms of PSNR, SSIM, run time, and complexity.

To comprehensively evaluate the SISR performance of the proposed HGSRCNN, we use the FSIM values of different methods to test their visual effects in terms of perception. Table X proves that the proposed HGSRCNN obtained the highest values than these of CFSRCNN and ACNet on B100 for three different scales (i.e., ×2, ×3, and ×4). According to mentioned illustrations, we can see that the proposed HGSRCNN is very effective in quantitative analysis for SISR.

2) Qualitative Analysis: To test the visual results of the proposed HGSRCNN, we choose six popular methods (i.e., VDSR, DRCN, CRAN-M, LESRCNN, CFSRCNN, and ACNet) on U100 and B100 to conduct predicted high-quality images. To easier observe detailed information of constructed SR images from different methods, one area of the predicted image is amplified as an observation area. The observation area is clearer, which implies its corresponding SR method has a better performance. Figs. 3 and 4 point out that the marked regions of the HGSRCNN are clearer than these of other SR methods. In other words, the proposed HGSRCNN outperforms other methods for SISR. According to the quantitative analysis and qualitative analysis, we can see that the proposed HGSRCNN is beneficial to SISR on digital devices.

This article has the following contributions.

1) The proposed 52-layer HGSRCNN uses heterogeneous architectures and RBs to enhance internal and external interactions of different channels both in parallel and serial ways for obtaining richer low-frequency structure information of different types, which is very suitable to SISR in complex scenes.
2) A multilevel enhancement mechanism guides a CNN to implement a symmetric architecture for progressively facilitating structural information in SISR.

3) The designed HGSRCNN obtains competitive execution speed for SISR. That is, it only takes the run time to 4.58% of RDN and 4.27% of SRFBN in restoring a high-quality image with 1024 × 1024.

V. Conclusion

In this article, we propose an HGSRCNN. The HGSR-CNN uses heterogeneous architectures in a parallel way to enhance internal and external relations of different channels for facilitating riches low-frequency structure information. Taking effects of obtained redundant features into consideration, an RB with signal enhancements in a serial way is conducted to filter useless information. To prevent the loss of original information, a multilevel enhancement mechanism guides a CNN to implement a symmetric architecture for promoting the expressive ability of HGSRCNN. Besides, a parallel upsampling mechanism is developed to train a blind SR model. A lot of experiments are conducted on four benchmark datasets to prove the effectiveness of the proposed HGSRCNN in terms of SISR results, SISR efficiency, complexity, and visual effects.

REFERENCES

[1] Z. Chen, X. Guo, P. Y. M. Woo, and Y. Yuan, “Super-resolution enhanced medical image diagnosis with sample affinity interaction,” IEEE Trans. Med. Imag., vol. 40, no. 5, pp. 1377–1389, May 2021.

[2] J. Shermeyer and A. Van Etten, “The effects of super-resolution on object detection performance in satellite imagery,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2019, pp. 1432–1441.

[3] K. Zhang, W. Zuo, and L. Zhang, “Learning a single convolutional super-resolution network for multiple degradations,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 3262–3271.

[4] M.-C. Chiang and T. E. Boult, “Efficient image warping and super-resolution,” in Proc. 3rd IEEE Workshop Appl. Comput. Vis. (WACV), Dec. 1996, pp. 56–61.

[5] V. K. Ha, J. Ren, X. Xu, S. Zhao, G. Xie, and V. M. Vargas, “Deep learning based single image super-resolution: A survey,” in Proc. Int. Conf. Brain Inspired Cognit. Syst. Cham, Switzerland: Springer, 2018, pp. 106–119.

[6] 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.

[7] 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.

[8] W. Yang, X. Zhang, Y. Tian, W. Wang, and J. Xue, “Deep learning for single image super-resolution: A brief review,” IEEE Trans. Multimedia, vol. 21, no. 12, pp. 3106–3121, May 2019.

[9] Z. Wang, J. Chen, and S. C. H. Hoi, “Deep learning for image super-resolution: A survey,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 10, pp. 3365–3387, Oct. 2021.

[10] C. Dong, C. C. Loy, K. He, and X. Tang, “Image super-resolution using deep convolutional networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 2, pp. 295–307, Feb. 2015.

[11] J. Kim, J. K. Lee, and K. M. Lee, “Accurate image super-resolution using very deep convolutional networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 1646–1654.

[12] J. Kim, J. K. Lee, and K. M. Lee, “Deeply-recursive convolutional network for image super-resolution,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 1637–1645.

[13] Y. Tai, J. Yang, and X. Liu, “Image super-resolution via deep recursive residual network,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jul. 2017, pp. 3147–3155.

[14] X. Mao, C. Shen, and Y.-B. Yang, “Image restoration using very deep convolutional encoder–decoder networks with symmetric skip connections,” in Proc. Adv. Neural Inf. Process. Syst., vol. 29, 2016, pp. 2802–2810.

[15] Y. Tai, J. Yang, X. Liu, and C. Xu, “MemNet: A persistent memory network for image restoration,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 4539–4547.

[16] C. Dong, C. C. Loy, and X. Tang, “Accelerating the super-resolution convolutional neural network,” in Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2016, pp. 391–407.

[17] Y. Huang et al., “Learning deformable and attentive network for image restoration,” Knowl.-Based Syst., vol. 231, Nov. 2021, Art. no. 107384.

[18] C. Tian et al., “Lightweight image super-resolution with enhanced RDN,” Knowl.-Based Syst., vol. 205, Oct. 2020, Art. no. 106235.

[19] Z. Li, J. Yang, Z. Liu, X. Yang, G. Jeon, and W. Wu, “Feedback network for image super-resolution,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 3867–3876.

[20] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, “Residual dense network for image super-resolution,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 2472–2481.

[21] Q. Wang, Q. Gao, L. Wu, G. Sun, and L. Liao, “Adversarial multi-path residual network for image super-resolution,” IEEE Trans. Image Process., vol. 30, pp. 6648–6658, 2021.

[22] N. Ahn, B. Kang, and K.-A. Sohn, “Fast, accurate, and lightweight super-resolution with cascading residual network,” in Proc. Eur. Conf. Comput. Vis. (ECCV), Sep. 2018, pp. 252–268.

[23] C. Tian, Y. Xu, W. Zuo, B. Zhang, L. Fei, and C.-W. Lin, “Coarse-to-fine CNN for image super-resolution,” IEEE Trans. Multimedia, vol. 23, pp. 1489–1502, 2021.

[24] X. Yang et al., “DRFN: Deep recurrent fusion network for single-image super-resolution with large factors,” IEEE Trans. Multimedia, vol. 21, no. 2, pp. 328–337, Feb. 2019.

[25] K. Prajapati et al., “Channel split convolutional neural network (ChaS-Net) for thermal image super-resolution,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2021, pp. 4368–4377.

[26] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, “Image super-resolution using very deep residual channel attention networks,” in Proc. Eur. Conf. Comput. Vis. (ECCV), Sep. 2018, pp. 286–301.

[27] B. Niu et al., “Single image super-resolution via a holistic attention network,” in Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2020, pp. 191–207.

[28] A. Yang, B. Yang, Z. Ji, Y. Pang, and L. Shao, “Lightweight group convolutional network for single image super-resolution,” Inf. Sci., vol. 516, pp. 220–233, Apr. 2020.

[29] V. Jain, P. Bansal, A. K. Singh, and R. Srivastava, “Efficient single image super resolution using enhanced learned group convolutions,” in Proc. Int. Conf. Neural Inf. Process. Cham, Switzerland: Springer, 2018, pp. 466–475.

[30] Z. Hui, X. Gao, Y. Yang, and X. Wang, “Lightweight image super-resolution with information multi-distillation network,” in Proc. 27th ACM Int. Conf. Multimedia, Oct. 2019, pp. 2024–2032.

[31] X. Zhao, Y. Zhang, T. Zhang, and X. Zou, “Channel splitting network for single MR image super-resolution,” IEEE Trans. Image Process., vol. 28, no. 11, pp. 5649–5662, Nov. 2019.

[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), vol. 25. Stateline, NV, USA, Dec. 2012, pp. 1097–1105.

[33] C. Douillard, M. Jézéquel, C. Berrou, A. Picart, P. Didier, and A. Glavieux, “Iterative correction of intersymbol interference: Turboequalization,” Eur. Trans. Telecommun., vol. 6, no. 5, pp. 507–511, 1995.

[34] E. Agustsson and R. Timofte, “NTIRE 2017 challenge on single image super-resolution: Dataset and study,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jul. 2017, pp. 126–135.

[35] M. Bevilacqua, A. Roumy, C. Guillotem, and M.-L.-A. Morel, “Low-complexity single-image super-resolution based on nonnegative neighbor embedding,” in Proc. Brit. Mach. Vis. Conf.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
[38] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, arXiv:1412.6980.

[39] J. Bergstra et al., “Theano: A CPU and GPU math compiler in Python,” in Proc. Python Sci. Conf., 2010, pp. 1–7.

[40] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, arXiv:1409.1556.

[41] A. Hore and D. Ziou, “Image quality metrics: PSNR vs. SSIM,” in Proc. 20th Int. Conf. Pattern Recognit., Aug. 2010, pp. 2366–2369.

[42] C. Szegedy et al., “Going deeper with convolutions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 1–9.

[43] J. Sun, Z. Xu, and H.-Y. Shum, “Image super-resolution using gradient profile prior,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2008, pp. 1–8.

[44] R. Timofte, V. De Smet, and L. Van Gool, “A+: Adjusted anchored neighborhood regression for fast super-resolution,” in Proc. Asian Conf. Comput. Vis. Cham, Switzerland: Springer, 2014, pp. 111–126.

[45] L. Zhang, L. Zhang, X. Mou, and D. Zhang, “FSIM: A feature similarity index for image quality assessment,” IEEE Trans. Image Process., vol. 39, no. 6, pp. 3718–3730, Jun. 2021.

[46] K. Jiang, Z. Wang, P. Yi, and J. Jiang, “Layered denoising network for image super-resolution,” Pattern Recognit., vol. 107, Nov. 2020, pp. 10601–10610.

[47] Y. Shi, K. Wang, C. Chen, L. Xu, and L. Lin, “Structure-preserving image super-resolution via contextualized multitask learning,” IEEE Trans. Multimedia, vol. 19, no. 12, pp. 2804–2815, Dec. 2017.

[48] Z. Hui, X. Wang, and X. Gao, “Fast and accurate single image super-resolution via information distillation network,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 723–731.

[49] Y. Fan et al., “Balanced two-stage residual networks for image super-resolution,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jul. 2017, pp. 161–168.

[50] Y. Wang, L. Wang, H. Wang, and P. Li, “End-to-end image super-resolution via deep and shallow convolutional networks,” IEEE Access, vol. 7, pp. 31959–31970, 2019.

[51] C. Tian, Y. Xu, W. Zuo, C.-W. Lin, and D. Zhang, “Asymmetric CNN for image superresolution,” IEEE Trans. Syst., Man, Cybern. Syst., vol. 52, no. 6, pp. 3718–3730, Jun. 2022.

[52] J. Jiang, K. Zhang, S. Gu, L. V. Gool, and R. Timofte, “Flow-based kernel prior with application to blind super-resolution,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 10611–10620.

[53] J. Liang, K. Zhang, S. Gu, L. V. Gool, and R. Timofte, “Flow-based kernel prior with application to blind super-resolution,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 10611–10620.

[54] Y. Shi, K. Wang, C. Chen, L. Xu, and L. Lin, “Structure-preserving image super-resolution via contextualized multitask learning,” IEEE Trans. Multimedia, vol. 19, no. 12, pp. 2804–2815, Dec. 2017.

[55] H. Ren, M. El-Khamy, and J. Lee, “Image super resolution based on fusing multiple convolution neural networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jul. 2017, pp. 54–61.

[56] W.-S. Lai, J.-B. Huang, N. Abuja, and M.-H. Yang, “Deep Laplacian pyramid networks for fast and accurate super-resolution,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 624–632.