Accurate and Efficient Single Image Super-Resolution with Matrix Channel Attention Network

Hailong Ma[0000−0002−1881−9650], Xiangxiang Chu[0000−0003−2548−0605], Bo Zhang[0000−0003−0564−617X]

Xiaomi AI Lab
{mahailong,chuxiangxiang,zhangbo11}@xiaomi.com

Abstract. In recent years, deep learning methods have achieved impressive results with higher peak signal-to-noise ratio in Single Image Super-Resolution (SISR) tasks. However, these methods are usually computationally expensive, which constrains their application in mobile scenarios. In addition, most of the existing methods rarely take full advantage of the intermediate features which are helpful for restoration. To address these issues, we propose a moderate-size SISR network named matrix channel attention network (MCAN) by constructing a matrix ensemble of multi-connected channel attention blocks (MCAB). Several models of different sizes are released to meet various practical requirements. Extensive benchmark experiments show that the proposed models achieve better performance with much fewer multiply-adds and parameters[1].

1 Introduction

Single image super-resolution (SISR) attempts to reconstruct a high-resolution (HR) image from its low-resolution (LR) counterpart, which is essentially an ill-posed inverse problem since there are infinitely many HR images that can be downsampled to the same LR image.

Most of the deep learning methods discussing SISR have been devoted to achieving higher *peak signal noise ratios* (PSNR) with deeper and deeper layers, making it difficult to fit in mobile devices [4, 3, 5, 8]. A lightweight framework CARN stands out for being one of the first SR methods that are applicable on mobile devices, however it comes at a cost of reduction on PSNR [7]. An information distillation network (IDN) also achieves good performance with a moderate model size [9]. Another effort that tackles SISR with neural architecture search has also been proposed [10], with their network FALSR surpassing CARN at the same level of FLOPs. Readers can find a thorough summary of state-of-the-art methods and results in [11].

[1] Source code is at https://github.com/macn3388/MCAN

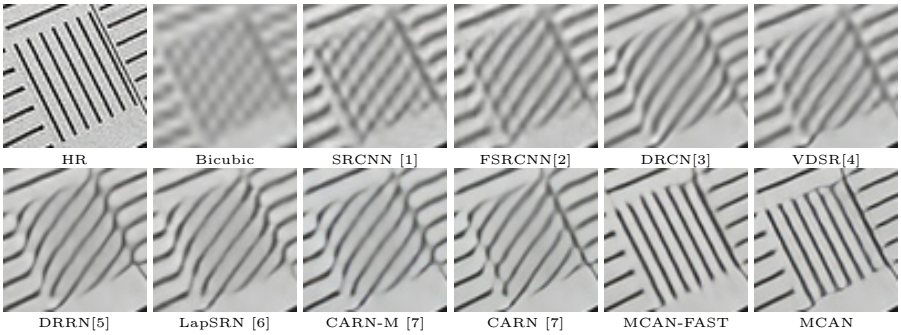

Fig. 1. Visual results with bicubic degradation (×4) on "img_092" from Urban100.

Nevertheless, it is worth pointing out that higher PSNR does not necessarily give better visual effects, for which a new measure called *perceptual index* (PI) is formulated [12]. Noteworthy works engaging perceptual performance are SRGAN [13] and ESRGAN [14], whose PSNR is not the highest, but both render more high-frequency details. However, these GAN-based methods inevitably bring about bad cases that are intolerable in practice. Our work still focuses on improving PSNR, which is a well-established distortion measure. Furthermore, our proposed model can also serve as the generator of GAN-based methods.

To seek a better trade-off between image quality and model sizes, we design an architecture called Matrix Channel Attention Network. We name its basic building block *multi-connected channel attention block* (MCAB), which is an adaptation of *residual channel attention block* (RCAB) from RCAN [8]. MCAB differs from RCAB by allowing hierarchical connections after each activation. In such a way, multiple levels of information can be passed both in depth and in breadth.

In summary, our main contributions are as follows:

- We propose a matrix channel attention network named MCAN for SISR. Our matrix-in-matrix (MIM) structure, which is composed of multi-connected channel attention blocks (MCAB), can effectively utilize the hierarchical features. We also devise a hierarchical feature fusion (HFF) block, which can be used in combination with the MIM structure. HFF can better profit the hierarchical features of MIM from the LR space.
- We build three additional efficient SR models of different sizes, namely MCAN-M, MCAN-S, MCAN-T, which correspond to the model sizes of mobile, small, and tiny, respectively. We also introduce MCAN-FAST by replacing the *sigmoid* with the *fast sigmoid* [15] to overcome the inefficiency of the sigmoid function on mobile devices, and MCAN-FAST has only a small loss of precision compared to MCAN.
- We show through extensive experiments that our MCAN family achieves higher PSNR and better perceptual results, with much fewer mult-adds and number of parameters than the state-of-the-art methods.

2 Related Work

In recent years, deep learning has been applied to many areas of computer vision [16–20]. Dong et al. [1] first applied deep learning to the image super-resolution field. They proposed a simple three-layer convolutional neural network called SRCNN, where each layer sequentially deals with feature extraction, non-linear mapping, and reconstruction. However, it needs an extra bicubic interpolation which reduces high-frequency information and adds extra computation. Their follow-up work FSRCNN [2] requires no interpolation and inserts a deconvolution layer for reconstruction, which learns an end-to-end mapping. Besides, shrinking and expanding layers are introduced to speed up computation, altogether rendering FSRCNN real-time on a generic CPU.

Meantime, VDSR presented by [3] proposes global residual learning to ease training for their very deep network. DRCN handles deep network recursively to share parameters [3]. DRRN builds two residual blocks in a recursive manner [5]. Unfortunately, these very deep architectures undoubtedly require heavy computation.

The application of DenseNet in SR domain goes to SRDenseNet [21], in which they argue that dense skip connections could mitigate the vanishing gradient problem and can boost feature propagation. It achieves better performance as well as faster speed. However, results from [10] showed that it is not efficient to connect all modules intensively because it brings extra computation, and their less dense network FALSR is also competitive.

A cascading residual network CARN is devised for a lightweight scenario [7]. The basic block of their architecture is called *cascading residual block*, whose outputs of intermediary layers are dispatched to each of the consequent convolutional layers. These cascading blocks, when stacked, are again organized in the same fashion.

Another remarkable work RCAN [8] observed that low-frequency information is hard to capture by convolutional layers which only exploit local regions. By adding multiple long and short skip connections for residual dense blocks, low-frequency features can bypass the network and thus the main architecture focuses on high-frequency information. RCAN also invented a *channel attention* mechanism via global average pooling to deal with interdependencies among channels.

3 Matrix Channel Attention Network

3.1 Network Structure

MCAN consists of four components: feature extraction (FE), a matrix-in-matrix (MIM) mapping, hierarchical feature fusion (HFF) and reconstruction, as illustrated in Figure 2.

Specifically, we utilize two successive 3×3 convolutions to extract features from the input image in the FE stage. Let I_{LR} represent the low-resolution

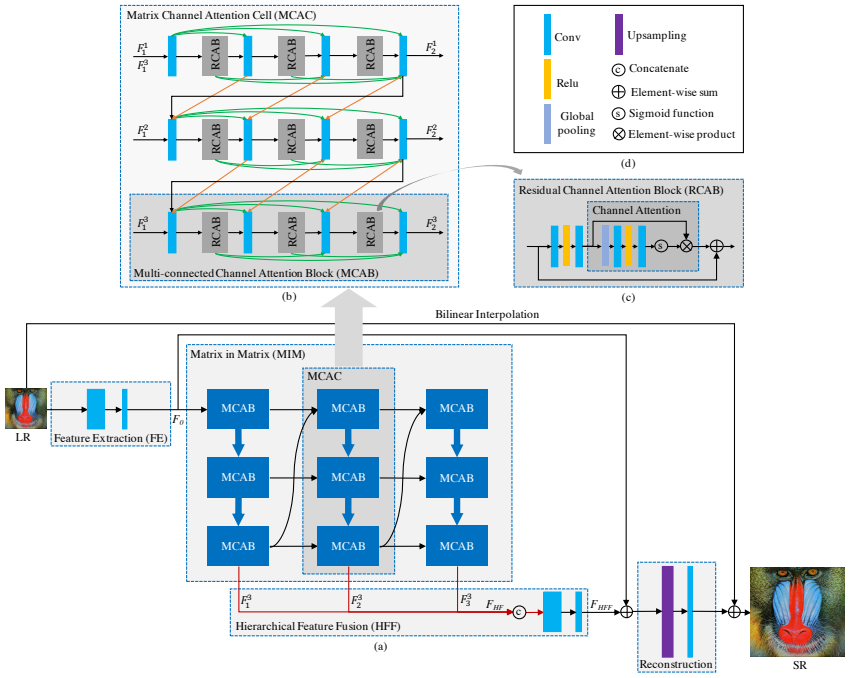

Fig. 2. (a) The overall structure of our matrix convolutional neural network (MCAN), in which MIM is set to $D = 3, K = 3, M = 3$. The blue thick arrows indicate multiple connections between two blocks. (b) Matrix channel attention cell (MCAC) composed by multi-connected channel attention block (MCAB). (c) Residual channel attention block proposed by RCAN [8].

input image and I_{HR} be the high-resolution output, this procedure can then be formulated as

$$F_0 = H_{FE}(I_{LR}) \tag{1}$$

where $H_{FE}(\cdot)$ is the feature extraction function and F_0 denotes the output features.

The nonlinear mapping is constructed by what we call a matrix-in-matrix module (MIM). Similarly,

$$F_{HF} = H_{MIM}(F_0) \tag{2}$$

where $H_{MIM}(\cdot)$ is the mapping function, to be discussed in detail in Section 3.2. F_{HF} stands for the hierarchical features, so named for they come from different depths of the our MIM structure. Further feature fusion can be put formally as

$$F_{HFF} = H_{HFF}(F_{HF}). \tag{3}$$

HFF will be elaborated in Section 3.4.

We then upscale the combination of fused feature F_{HFF} and F_0, and add it to the interpolated I_{LR} to generate the high-resolution target,

$$I_{SR} = H_{UP}(F_{HFF} + F_0) + B(I_{LR}), \tag{4}$$

where $H_{UP}(\cdot)$ denotes the upsampling function and $B(\cdot)$ the bilinear interpolation.

3.2 Matrix in Matrix

In general, an MIM is composed of D matrix channel attention cells (MCAC). Each MCAC contains K MCABs, and each MCAB contains M residual channel attention blocks (RCAB) [8] and $M + 1$ pointwise convolution layers.

As shown in Figure 2(a), MCACs in the MIM are firstly connected from end to end, marked by black curved arrows. MCABs inside an MCAC also contributes for connections, which are evinced by black straight arrows.

Likewise, MCABs within an MCAC are also multi-connected, indicated by blue thick arrows. Figure 2(b) shows its details: MCABs are joined one after another, which are represented by black polyline arrows. Meanwhile, orange arrows add a bridge between the pointwise convolution layers and its equivalent in the next MCAB.

Therefore, an MCAC can be seen as a matrix of size $K \times M$. An MIM containing D MCACs can be thus regarded as a $D \times K \times M$ matrix, for which reason we nicknamed the structure "matrix-in-matrix". Figure 2 gives an example of an MIM of $3 \times 3 \times 3$ matrix, containing 3 MCACs, and each MCAC has 3 MCABs. We will elaborate MCABs in Section 3.3.

3.3 Matrix Channel Attention Cell

In super-resolution, skip connections are popular since it reuses intermediate features while relieving the training for deep networks [3, 22, 21]. Nevertheless, these skip connections between modules are point-to-point, where only the output features of a module can be reused, losing many intermediate features. This can be alleviated by adding skip connections within the module, but as more intermediate features are concatenated, channels become very thick before fusion [23], which narrows transmission of information and gradients.

If we densely connect all intermediate features and modules like SRDenseNet [21], it inevitably brings in redundant connections for less important features, while the important ones become indistinguishable, which increases the training difficulty.

To address these problems, we propose a matrix channel attention cell, which is composed of several multi-connected channel attention blocks. We recursively define F_d as the outputs of an MCAC,

$$\begin{aligned} F_d &= H^d_{MCAC}(F_{d-1}) \\ &= H^d_{MCAC}((F^1_{d-1}, F^2_{d-1}, \ldots, F^K_{d-1})) \\ &= (F^1_d, F^2_d, \ldots, F^K_d). \end{aligned} \tag{5}$$

Thence the output of $H_{MIM}(\cdot)$ can be composed by the combination of K-th outputs of all MCACs,

$$F_{EF} = (F_1^K, F_2^K, \ldots, F_D^K). \tag{6}$$

Multi-connected Channel Attention Block Previous works seldom discriminate feature channels and treat them equally. Till recently a channel attention mechanism using global pooling is proposed in RCAN to concentrate on more useful channels [8]. We adopt the same channel attention block RCAB as in RCAN, also depicted in Figure 2(c), and the difference between the two only lies in the style of connections.

Channel Attention Mechanism. We let $X = [x_1, \cdots, x_c, \cdots, x_C]$ denote an input that contains C feature maps, and the shape of each feature map be $H \times W$. Then the statistic z_c of the c-th feature map x_c is defined as

$$z_c = H_{GP}(x_c) = \frac{\sum_{i=1}^{H} \sum_{j=1}^{W} x_c(i,j)}{H \times W}, \tag{7}$$

where $x_c(i,j)$ denotes the value at index (i,j) of feature map x_c, and $H_{GP}(\cdot)$ represents the global average pooling function. The channel attention of the feature map x_c can thus be denoted as

$$s_c = f(W_U \delta(W_D z_c)), \tag{8}$$

where $f(\cdot)$ and $\delta(\cdot)$ represent the sigmoid function and the ReLU [24] function respectively, W_D is the weight set of a 1×1 convolution for channel downscaling. This convolution reduces the number of channels by a factor r. Later after being activated by a ReLU function, it enters a 1×1 convolution for channel upscaling with the weights W_U, which expands the channel again by the factor r. The computed statistic s_c is used to rescale the input features x_c,

$$\hat{x}_c = s_c \cdot x_c, \tag{9}$$

Description of RCAB. The RCAB is organized using the aforementioned channel attention mechanism. Formally it is a function $H_{RCAB}(\cdot)$ on the input features I,

$$\begin{aligned} F_{RCAB} &= H_{RCAB}(I) \\ &= s_{X_I} \cdot X_I + I \\ &= \hat{X}_I + I, \end{aligned} \tag{10}$$

where s_{X_I} is the output of channel attention on X_I, which are the features generated from the two stacked convolution layers,

$$X_I = W_2 \delta(W_1 I). \tag{11}$$

The cascading mechanism from CARN [7] makes use of intermediate features in a dense way. In order to relax the redundancy of dense skip connections,

our channel attention blocks are built in a multi-connected fashion, so-called as MCAB. As shown in Figure 2(b), each MCAB contains M residual channel attention blocks (RCAB) and $M+1$ pointwise convolution operations for feature fusion (H_F), which are interleaved by turns. In addition to the cascading mechanism marked by green arrows, we added multi-connections between MCABs, which are marked by orange arrows.

MCAC Structure As we mentioned before, each MCAC contains K MCABs and each MCAB is composed of M RCABs. In the d-th MCAC, we let the input and output of k-th MCAB be IM_d^k and OM_d^k, and the k-th output of the last MCAC be F_{d-1}^k, we formulate IM_d^k as follows,

$$\begin{cases} [F_0] & d=k=0 \\ [OM_d^{k-1}] & d=0, k\in(0,K] \\ [F_{d-1}^k] & d\in(0,D], k=0 \\ [OM_d^{k-1}, F_{d-1}^k] & d\in(0,D], k\in(0,K] \end{cases} \tag{12}$$

In the case of $d\in(0,D], k\in(0,K], m\in(0,M]$, the m-th feature fusion convolution H_F takes multiple inputs and fuses them into $F_d^{k,m}$. Let the input of m-th RCAB be $IR_d^{k,m}$ and the output $OR_d^{k,m}$, we can write the input of m-th feature fusion convolution $IF_d^{k,m}$ as

$$\begin{cases} [F_{d-1}^k, F_d^{k-1,m+1}, F_d^{k-1,M+1}] & m=0 \\ [OR_d^{k,m-1}, F_d^{k-1,m+1}, F_d^{k,1}, & \\ \quad \ldots, F_d^{k,m-1}] & m\in(0,M] \\ [OR_d^{k,m-1}, F_d^{k,1}, \ldots, F_d^{k,M}] & m=M+1 \end{cases} \tag{13}$$

Now we give the complete definition of the output of d-th MCAC,

$$\begin{aligned} F_d &= (F_d^1, F_d^2, \ldots, F_d^K) \\ &= H_{MCAC,d}(F_{d-1}) \\ &= H_{MCAC,d}((F_{d-1}^1, F_{d-1}^2, \ldots, F_{d-1}^K)) \\ &= (F_d^{1,M+1}, F_d^{2,M+1}, \ldots, F_d^{K,M+1}). \end{aligned} \tag{14}$$

As we have seen, the nonlinear mapping module of our proposed model is a matrix of $D\times K$. Thus its overall number of sigmoid functions can be calculated as

$$num_{f(\cdot)} = D\times K\times M\times N_{ca}, \tag{15}$$

where $f(\cdot)$ means the sigmoid function and N_{ca} indicates the number of filters in the channel attention mechanism.

Table 1. Quantitative comparison with the state-of-the-art methods based on ×2, ×3, ×4 (sequentially splitted in three sets of rows in the table) SR with a bicubic degradation mode. Red/blue text: best/second-best.

Method	Mult-Adds	Params	Set5	Set14	B100	Urban100
			PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
SRCNN [1]	52.7G	57K	36.66/0.9542	32.42/0.9063	31.36/0.8879	29.50/0.8946
FSRCNN [2]	6.0G	12K	37.00/0.9558	32.63/0.9088	31.53/0.8920	29.88/0.9020
VDSR [4]	612.6G	665K	37.53/0.9587	33.03/0.9124	31.90/0.8960	30.76/0.9140
DRCN [3]	17,974.3G	1,774K	37.63/0.9588	33.04/0.9118	31.85/0.8942	30.75/0.9133
LapSRN [6]	29.9G	813K	37.52/0.9590	33.08/0.9130	31.80/0.8950	30.41/0.9100
DRRN [5]	6,796.9G	297K	37.74/0.9591	33.23/0.9136	32.05/0.8973	31.23/0.9188
BTSRN [25]	207.7G	410K	37.75/-	33.20/-	32.05/-	31.63/-
MemNet [26]	2,662.4G	677K	37.78/0.9597	33.28/0.9142	32.08/0.8978	31.31/0.9195
SelNet [27]	225.7G	974K	37.89/0.9598	33.61/0.9160	32.08/0.8984	-
CARN [7]	222.8G	1,592K	37.76/0.9590	33.52/0.9166	32.09/0.8978	31.92/0.9256
CARN-M [7]	91.2G	412K	37.53/0.9583	33.26/0.9141	31.92/0.8960	31.23/0.9194
MoreMNAS-A [28]	238.6G	1,039K	37.63/0.9584	33.23/0.9138	31.95/0.8961	31.24/0.9187
FALSR-A [10]	234.7G	1,021K	37.82/0.9595	33.55/0.9168	32.12/0.8987	31.93/0.9256
MCAN	191.3G	1,233K	37.91/0.9597	33.69/0.9183	32.18/0.8994	32.46/0.9303
MCAN+	191.3G	1,233K	38.10/0.9601	33.83/0.9197	32.27/0.9001	32.68/0.9319
MCAN-FAST	191.3G	1,233K	37.84/0.9594	33.67/0.9188	32.16/0.8993	32.36/0.9300
MCAN-FAST+	191.3G	1,233K	38.05/0.9600	33.78/0.9196	32.26/0.8999	32.62/0.9317
MCAN-M	105.50G	594K	37.78/0.9592	33.53/0.9174	32.10/0.8984	32.14/0.9271
MCAN-M+	105.50G	594K	37.98/0.9597	33.68/0.9186	32.200.8992	32.35/0.9290
MCAN-S	46.09G	243K	37.62/0.9586	33.35/0.9156	32.02/0.8976	31.83/0.9244
MCAN-S+	46.09G	243K	37.82/0.9592	33.49/0.9168	32.12/0.8983	32.03/0.9262
MCAN-T	6.27G	35K	37.24/0.9571	32.97/0.9112	31.74/0.8939	30.62/0.9120
MCAN-T+	6.27G	35K	37.45/0.9578	33.07/0.9121	31.85/0.8950	30.79/0.9137
SRCNN [1]	52.7G	57K	32.75/0.9090	29.28/0.8209	28.41/0.7863	26.24/0.7989
FSRCNN [2]	5.0G	12K	33.16/0.9140	29.43/0.8242	28.53/0.7910	26.43/0.8080
VDSR [4]	612.6G	665K	33.66/0.9213	29.77/0.8314	28.82/0.7976	27.14/0.8279
DRCN [3]	17,974.3G	1,774K	33.82/0.9226	29.76/0.8311	28.80/0.7963	27.15/0.8276
DRRN [5]	6,796.9G	297K	34.03/0.9244	29.96/0.8349	28.95/0.8004	27.53/0.8378
BTSRN [25]	207.7G	410K	37.75/-	33.20/-	32.05/-	31.63/-
MemNet [26]	2,662.4G	677K	34.09/0.9248	30.00/0.8350	28.96/0.8001	27.56/0.8376
SelNet [27]	120.0G	1,159K	34.27/0.9257	30.30/0.8399	28.97/0.8025	-
CARN [7]	118.8G	1,592K	34.29/0.9255	30.29/0.8407	29.06/0.8034	28.06/0.8493
CARN-M [7]	46.1G	412K	33.99/0.9236	30.08/0.8367	28.91/0.8000	27.55/0.8385
MCAN	95.4G	1,233K	34.45/0.9271	30.43/0.8433	29.14/0.8060	28.47/0.8580
MCAN+	95.4G	1,233K	34.62/0.9280	30.50/0.8442	29.21/0.8070	28.65/0.8605
MCAN-FAST	95.4G	1,233K	34.41/0.9268	30.40/0.8431	29.12/0.8055	28.41/0.8568
MCAN-FAST+	95.4G	1,233K	34.54/0.9276	30.48/0.8440	29.20/0.8067	28.60/0.8595
MCAN-M	50.91G	594K	34.35/0.9261	30.33/0.8417	29.06/0.8041	28.22/0.8525
MCAN-M+	50.91G	594K	34.50/0.9271	30.44/0.8432	29.14/0.8053	28.39/0.8552
MCAN-S	21.91G	243K	34.12/0.9243	30.22/0.8391	28.99/0.8021	27.94/0.8465
MCAN-S+	21.91G	243K	34.28/0.9255	30.31/0.8403	29.07/0.8034	28.09/0.8493
MCAN-T	3.10G	35K	33.54/0.9191	29.76/0.8301	28.73/0.7949	26.97/0.8243
MCAN-T+	3.10G	35K	33.68/0.9207	29.8/0.8320	28.80/0.7964	27.10/0.8271
SRCNN [1]	52.7G	57K	30.48/0.8628	27.49/0.7503	26.90/0.7101	24.52/0.7221
FSRCNN [2]	4.6G	12K	30.71/0.8657	27.59/0.7535	26.98/0.7150	24.62/0.7280
VDSR [4]	612.6G	665K	31.35/0.8838	28.01/0.7674	27.29/0.7251	25.18/0.7524
DRCN [3]	17,974.3G	1,774K	31.53/0.8854	28.02/0.7670	27.23/0.7233	25.14/0.7510
LapSRN [6]	149.4G	813K	31.54/0.8850	28.19/0.7720	27.32/0.7280	25.21/0.7560
DRRN [5]	6,796.9G	297K	31.68/0.8888	28.21/0.7720	27.38/0.7284	25.44/0.7638
BTSRN [25]	207.7G	410K	37.75/-	33.20/-	32.05/-	31.63/-
MemNet [26]	2,662.4G	677K	31.74/0.8893	28.26/0.7723	27.40/0.7281	25.50/0.7630
SelNet [27]	83.1G	1,417K	32.00/0.8931	28.49/0.7783	27.44/0.7325	-
SRDenseNet [21]	389.9G	2,015K	32.02/0.8934	28.50/0.7782	27.53/0.7337	26.05/0.7819
CARN [7]	90.9G	1,592K	32.13/0.8937	28.60/0.7806	27.58/0.7349	26.07/0.7837
CARN-M [7]	32.5G	412K	31.92/0.8903	28.42/0.7762	27.44/0.7304	25.62/0.7694
CARN1 [29]	11.3G	86.24K	31.13/0.88	27.93/0.76	27.20/0.72	25.05/0.74
OISR [30]	114.2G	1.52M	32.14/0.8947	28.63/0.7819	27.60/0.7369	26.17/0.7888
IMDN [31]	71.99G	715K	32.21/0.8948	28.58/0.7811	27.56/0.7353	26.04/0.7838
MCAN	83.1G	1,233K	32.33/0.8959	28.72/0.7835	27.63/0.7378	26.43/0.7953
MCAN+	83.1G	1,233K	32.48/0.8974	28.80/0.7848	27.69/0.7389	26.58/0.7981
MCAN-FAST	83.1G	1,233K	32.30/0.8955	28.69/0.7829	27.60/0.7372	26.37/0.7938
MCAN-FAST+	83.1G	1,233K	32.43/0.8970	28.78/0.7843	27.68/0.7385	26.53/0.7970
MCAN-M	35.53G	594K	32.21/0.8946	28.63/0.7813	27.57/0.7357	26.19/0.7877
MCAN-M+	35.53G	594K	32.34/0.8959	28.72/ 0.7827	27.63/0.7370	26.34/0.7909
MCAN-S	13.98G	243K	31.97/0.8914	28.48/0.7775	27.48/0.7324	25.93/0.7789
MCAN-S+	13.98G	243K	32.11/0.8932	28.57/0.7791	27.55/0.7338	26.06/0.7822
MCAN-T	2.00G	35K	31.33/0.8812	28.04/0.7669	27.22/0.7228	25.12/0.7515
MCAN-T+	2.00G	35K	31.50/0.8843	28.14/0.7689	27.29/0.7244	25.23/0.7548

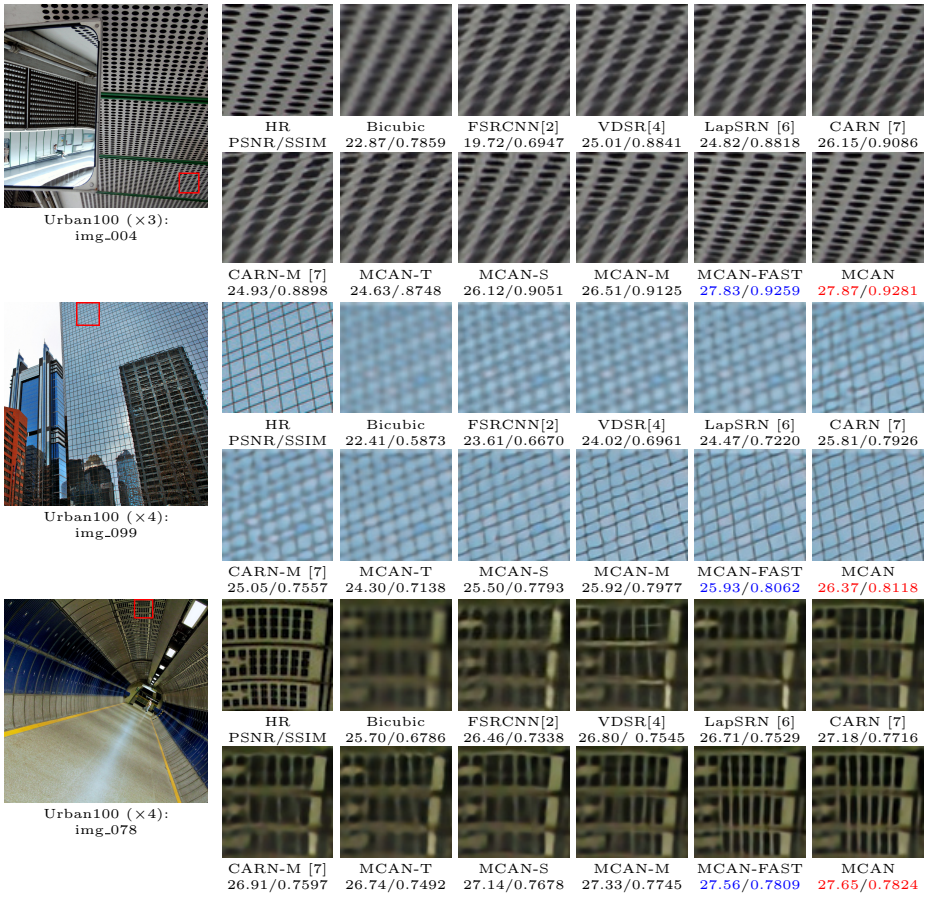

Fig. 3. Visual comparison with bicubic degradation model. Red/blue text: best/second-best.

3.4 Hierarchical Feature Fusion

Since we generate multiple features through MIM during different stages, we put forward a hierarchical feature fusion (HFF) module to integrate these features hierarchically.

Particularly, we unite the outputs of the last MCAB in each MCAC as the hierarchical features of MIM, which are marked by red arrows in Figure 2(a). In further detail, HFF takes a 3×3 convolution for fusion and another 3×3 convolution to reduce channel numbers:

$$F_{HFF} = W_F W_R[F_d^{1,M+1}, \ldots, F_d^{K,M+1}], \tag{16}$$

where W_F and W_R are the weights of fusion convolution and the channel reduction layer.

3.5 Comparison with Recent Models

Comparison with SRDenseNet. SRDenseNet uses dense blocks proposed by DenseNet to construct a nonlinear mapping module [21]. This dense connection mechanism may lead to redundancy, in fact, not all features should be equally treated. In our work, MIM and HFF can reduce dense connections and highlight the hierarchical information. Additionally, SRDenseNet connects two blocks from point to point, which refrains transmission and utilization of intermediate features. Our proposed multi-connected channel attention block (MCAB) mitigates this problem by injecting multiple connections between blocks.

Comparison with CARN. CARN uses a cascading mechanism [7], which is also pictured in our MIM. Despite this, MIM features multiple connections between MCACs, and the outputs of different stages are relayed between MCABs. Such an arrangement makes better use of intermediate information. Another important difference is that MCAN combines the hierarchical features before upsampling via hierarchical feature fusion. This mechanism helps significantly for reconstruction.

4 Experimental Results

4.1 Datasets and Evaluation Metrics

We train our model based on DIV2K [32], which contains 800 2K high-resolution images for the training set and another 100 pictures for both the validation and test set. Besides, we make comparisons across three scaling tasks ($\times 2$, $\times 3$, $\times 4$) on four datasets: Set5 [33], Set14 [34], B100 [35], and Urban100 [36]. The evaluation metrics we used are PSNR [37] and SSIM [38] on the Y channel in the YCbCr space.

4.2 Implementation Details

As shown in Figure 2, the inputs and outputs of our model are RGB images. We crop the LR patches by 64×64 for various scale tasks and adopt the standard data augmentation.

For training, we use Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$) [39] to minimize L_1 loss within 1.2×10^6 steps with a batch-size of 64. The initial learning rate is set to 2×10^{-4}, halved every 4×10^5 steps. Like CARN [7], we also initialize the network parameters by $\theta \sim U(-k, k)$, where $k = 1/\sqrt{c_{in}}$ and c_{in} is the number of input feature maps. Inspired by EDSR [40], we apply a multi-scale training. Our sub-pixel convolution is the same as in ESPCN [41].

We choose network hyperparameters to build an accurate and efficient model. The first two layers in the FE stage contain $n_{FE} = \{64, 32\}$ filters accordingly. As for MIM, we set $D = K = M = 3$, its number of filters $n_{MIM} = 32$. Two HFF convolutions have $n_{HFF} = \{D \times 32, 32\}$ filters. The last convolution before the upsampling procedure has $n_l = 256$ filters. The reduction factor r in the channel attention mechanism is set to 8.

Table 2. Hyperparameters of our networks.

Models	n_{FE}	n_{MIM}	n_{HFF}	n_l	r
MCAN	{64,32}	32	{96,32}	256	8
MCAN-FAST	{64,32}	32	{96,32}	256	8
MCAN-M	{64,24}	24	{72,24}	128	8
MCAN-S	{32,16}	16	{48,16}	64	8
MCAN-T	{16,8}	8	{24,8}	24	4

Since the *sigmoid* function is inefficient on some mobile devices, especially for some fixed point units such as DSPs. Therefore we propose MCAN-FAST by replacing the *sigmoid* with the *fast sigmoid* [15], which can be written as,

$$f_{fast}(x) = \frac{x}{1+|x|}. \tag{17}$$

Experiments show that MCAN-FAST has only a small loss on precision, and it achieves almost the same level of metrics as MCAN.

For more lightweight applications, we reduce the number of filters as shown in Table 4.2. Note in MCAN-T we also set the group as 4 in the group convolution of RCAB for further compression.

4.3 Comparisons with State-of-the-art Algorithms

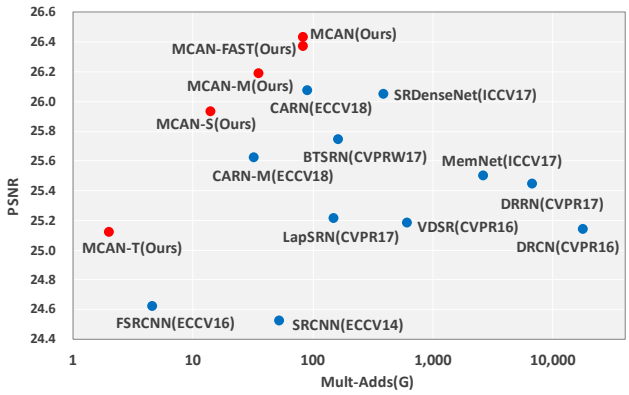

Fig. 4. MCAN models (red) compared with others (blue) on ×4 task of Urban 100. The mult-adds are calculated when the target HR image has a size of 1280×720.

We use Mult-Adds and the number of parameters to measure the model size. We emphasize on Mult-Adds as it indicates the number of multiply-accumulate operations. By convention, it is normalized on 1280×720 high-resolution images.

Further evaluation based on geometric self-ensemble strategy [40] are marked with '+'.

Quantitative comparisons with the state-of-the-art methods are listed in Table 3.3. For fair comparison, we concentrate on models with comparative mult-adds and parameters.

Notably, MCAN outperforms CARN [7] with fewer mult-adds and parameters. The medium-size model MCAN-M [7] achieves better performance than the CARN-M, additionally, it is still on par with CARN with about half of its mult-adds. For short, it surpasses all other listed methods, including MoreMNAS-A [28] and FALSR-A [10] from NAS methods.

The smaller model MCAN-S emulates LapSRN [6] with much fewer parameters. Particularly, it has an average advantage of 0.5 dB on PSNR over the LapSRN on the $\times 2$ task, and on average, MCAN-S still has an advantage of 0.4 dB. MCAN-S also behaves better than CARN-M on all tasks with half of its model size. It is worth to note that heavily compressed MCAN-S still exceeds or matches larger models such as VDSR, DRCN, DRRN, and MemNet.

The tiny model MCAN-T is meant to be applied under requirements of extreme fast speed. It overtakes FSRCNN [2] on all tasks with the same level of mult-adds.

4.4 Ablation Study

In this section, we demonstrate the effectiveness of the MIM structure and HFF through ablation study.

Matrix in matrix. We remove the connections between MCACs and also the connections between MCABs. Hence the model comes without intermediate connections. As shown in Table 3, the MIM structure can bring significant improvements, PSNR improves from 29.44 dB to 30.25 dB when such connections are enabled. When HFF is added, PNSR continues to increase from 30.23 dB to 30.28 dB.

Table 3. Investigation of MIM and HFF. We record the best average PSNR(dB) values of Set5 & Set14 on $\times 4$ SR task in 10^5 steps.

MIM	✘	✔	✘	✔
HFF	✘	✘	✔	✔
Avg. PSNR	29.44	30.25	30.23	30.28

Hierarchical feature fusion. We simply eliminate the fusion convolutions connected to MIM and consider the output of the last MCAB as the output of MIM. In this case, the intermediate features acquired by the MIM structure are not directly involved in the reconstruction. In Table 3, we observe that the HFF structure enhances PNSR from 29.44 dB to 30.23 dB. With MIM enabled, PSNR is further promoted from 30.25 dB to 30.28 dB.

5 Conclusion

In this paper, we proposed an accurate and efficient network with matrix channel attention for the SISR task. Our main idea is to exploit the intermediate features hierarchically through multi-connected channel attention blocks. MCABs then act as a basic unit that builds up the matrix-in-matrix module. We release three additional efficient models of varied sizes, MCAN-M, MCAN-S, and MCAN-T. Extensive experiments reveal that our MCAN family excel the state-of-the-art models of accordingly similar or even bigger sizes.

To deal with the inefficiency of the sigmoid function on some mobile devices, we benefit from the fast sigmoid to construct MCAN-FAST. The result confirms that MCAN-FAST has only a small loss of precision when compared to MCAN, and it can still achieve better performance with fewer mult-adds and parameters than the state-of-the-art methods.

References

1. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV. (2014) 184–199
2. Dong, C., Loy, C.C., Tang, X.: Accelerating the super-resolution convolutional neural network. In: Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II. (2016) 391–407
3. Kim, J., Lee, J.K., Lee, K.M.: Deeply-recursive convolutional network for image super-resolution. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. (2016) 1637–1645
4. Kim, J., Lee, J.K., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. (2016) 1646–1654
5. Tai, Y., Yang, J., Liu, X.: Image super-resolution via deep recursive residual network. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. (2017) 2790–2798
6. Lai, W., Huang, J., Ahuja, N., Yang, M.: Deep laplacian pyramid networks for fast and accurate super-resolution. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. (2017) 5835–5843
7. Ahn, N., Kang, B., Sohn, K.: Fast, accurate, and lightweight super-resolution with cascading residual network. In: Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part X. (2018) 256–272
8. Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y.: Image super-resolution using very deep residual channel attention networks. In: Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII. (2018) 294–310

9. Hui, Z., Wang, X., Gao, X.: Fast and accurate single image super-resolution via information distillation network. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. (2018) 723–731
10. Chu, X., Zhang, B., Ma, H., Xu, R., Li, J., Li, Q.: Fast, accurate and lightweight super-resolution with neural architecture search. CoRR **abs/1901.07261** (2019)
11. Zhang, K., Gu, S., Timofte, R., Hui, Z., Wang, X., Gao, X., Xiong, D., Liu, S., Gang, R., Nan, N., et al.: AIM 2019 challenge on constrained super-resolution: Methods and results. In: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), IEEE (2019) 3565–3574
12. Blau, Y., Mechrez, R., Timofte, R., Michaeli, T., Zelnik-Manor, L.: The 2018 PIRM challenge on perceptual image super-resolution. In: Computer Vision - ECCV 2018 Workshops - Munich, Germany, September 8-14, 2018, Proceedings, Part V. (2018) 334–355
13. Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A.P., Tejani, A., Totz, J., Wang, Z., Shi, W.: Photo-realistic single image super-resolution using a generative adversarial network. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. (2017) 105–114
14. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., Loy, C.C.: ESRGAN: enhanced super-resolution generative adversarial networks. In: Computer Vision - ECCV 2018 Workshops - Munich, Germany, September 8-14, 2018, Proceedings, Part V. (2018) 63–79
15. Georgiou, G.: Parallel Distributed Processing in the Complex Domain. PhD thesis, Tulane (1992)
16. Girshick, R.B.: Fast R-CNN. In: 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015. (2015) 1440–1448
17. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017. (2017) 2980–2988
18. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S.E., Fu, C., Berg, A.C.: SSD: single shot multibox detector. In: Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I. (2016) 21–37
19. Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. In: 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015. (2015) 1520–1528
20. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III. (2016) 649–666
21. Tong, T., Li, G., Liu, X., Gao, Q.: Image super-resolution using dense skip connections. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017. (2017) 4809–4817
22. Mao, X., Shen, C., Yang, Y.: Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In: Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain. (2016) 2802–2810
23. Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y.: Residual dense network for image super-resolution. In: 2018 IEEE Conference on Computer Vision and Pattern

Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. (2018) 2472–2481
24. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel. (2010) 807–814
25. Fan, Y., Shi, H., Yu, J., Liu, D., Han, W., Yu, H., Wang, Z., Wang, X., Huang, T.S.: Balanced two-stage residual networks for image super-resolution. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2017, Honolulu, HI, USA, July 21-26, 2017. (2017) 1157–1164
26. Tai, Y., Yang, J., Liu, X., Xu, C.: Memnet: A persistent memory network for image restoration. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017. (2017) 4549–4557
27. Choi, J., Kim, M.: A deep convolutional neural network with selection units for super-resolution. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2017, Honolulu, HI, USA, July 21-26, 2017. (2017) 1150–1156
28. Chu, X., Zhang, B., Xu, R., Ma, H.: Multi-objective reinforced evolution in mobile neural architecture search. CoRR **abs/1901.01074** (2019)
29. Li, Y., Agustsson, E., Gu, S., Timofte, R., Van Gool, L.: Carn: Convolutional anchored regression network for fast and accurate single image super-resolution. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 0–0
30. He, X., Mo, Z., Wang, P., Liu, Y., Yang, M., Cheng, J.: Ode-inspired network design for single image super-resolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 1732–1741
31. Hui, Z., Gao, X., Yang, Y., Wang, X.: Lightweight image super-resolution with information multi-distillation network. In: Proceedings of the 27th ACM International Conference on Multimedia (ACM MM). (2019) 2024–2032
32. Agustsson, E., Timofte, R.: NTIRE 2017 challenge on single image super-resolution: Dataset and study. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2017, Honolulu, HI, USA, July 21-26, 2017. (2017) 1122–1131
33. Bevilacqua, M., Roumy, A., Guillemot, C., Alberi-Morel, M.: Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In: British Machine Vision Conference, BMVC 2012, Surrey, UK, September 3-7, 2012. (2012) 1–10
34. Yang, J., Wright, J., Huang, T.S., Ma, Y.: Image super-resolution via sparse representation. IEEE Trans. Image Processing **19** (2010) 2861–2873
35. Martin, D.R., Fowlkes, C.C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: ICCV. (2001) 416–425
36. Huang, J., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015. (2015) 5197–5206
37. Horé, A., Ziou, D.: Image quality metrics: PSNR vs. SSIM. In: 20th International Conference on Pattern Recognition, ICPR 2010, Istanbul, Turkey, 23-26 August 2010. (2010) 2366–2369
38. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Processing **13** (2004) 600–612

39. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR **abs/1412.6980** (2014)
40. Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single image super-resolution. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2017, Honolulu, HI, USA, July 21-26, 2017. (2017) 1132–1140
41. Shi, W., Caballero, J., Huszar, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. (2016) 1874–1883