LIGHTWEIGHT SINGLE IMAGE SUPER-RESOLUTION THROUGH EFFICIENT SECOND-ORDER ATTENTION SPINDLE NETWORK

Yiyun Chen[1], *Yihong Chen*[2], *Jing-Hao Xue*[3], *Wenming Yang*[1,*], *Qingmin Liao*[1]

[1]Shenzhen International Graduate School/Department of Electronic Engineering, Tsinghua University, China
[2]Department of Computer Science, University College London, UK
[3]Department of Statistical Science, University College London, UK
chenyy18@mails.tsinghua.edu.cn, yihong.chen.19@ucl.ac.uk
jinghao.xue@ucl.ac.uk, yangelwm@163.com, liaoqm@tsinghua.edu.cn

ABSTRACT

Recent years have witnessed great success of applying deep convolutional neural networks (CNNs) to single image super-resolution (SISR). However, most of these algorithms focus on increasing modeling capability through developing deeper and wider networks, improving the performance but at a cost of huge computation. Targeting at a better trade-off between efficiency and effectiveness, we propose ESASN, an efficient second-order attention spindle network for lightweight SISR. ESASN is built upon efficient second-order attention spindle (ESAS) blocks, each of which contains two well-designed new modules, efficient multi-scale (EMS) module and second-order attention (SOA) module. EMS reduces a considerable number of parameters while retaining the multi-scale structure to explore rich features. SOA further rescales the multi-scale feature maps, capturing the inter-dependencies among channels pixel-wisely with little additional cost. Both qualitative and quantitative experimental results demonstrate that the combination of EMS and SOA works out favorably for SISR, lifting the performance with fewer parameters. Code is available at *https://github.com/yiyunchen/ESASN*.

Index Terms— Lightweight super-resolution, multi-scale features, spindle network, second-order attention

1. INTRODUCTION

Single image super-resolution (SISR) is a fundamental computer vision task, aiming at restoring the high resolution (HR) image from a single low resolution (LR) image. It is widely used in real-world applications, such as surveillance and medical imaging. However, SISR is also a challenging ill-posed problem notoriously hard to solve.

Since Dong *et al.* [1] introduced a three-layer convolution neural network (SRCNN) for SISR, convolution neural network (CNN)-based models, characterized by their strong representation learning capability, have attracted considerable attention [2, 3, 4, 5, 6, 7, 8, 9]. A trend of designing such CNN-based models is to deepen and widen the CNN architecture for stronger modeling capacity. Although deeper and wider networks indeed can often achieve better results, they inevitably lead to an upsurge of model parameters, bringing in unaffordable computational cost and thus infeasible for low-resource scenarios in real-world applications.

To address this issue, some lightweight networks [10, 11, 12, 3] have been proposed. DRRN [3] employed a recursive parameter sharing strategy, reducing the number of parameters but not necessarily the computation cost due to its recursive nature. CARN-M [10] adopted group convolution to lower the number of parameters. However, simply applying group convolution may result in information interchange difficulty among different groups and thus drastically degrade performance. In LFFN [11], a well-designed block, named spindle, was proposed to efficiently improve the representation capability of the model by incorporating multi-scale features. Unlike CARN-M, LFFN incorporated multi-scale features by first slicing the feature maps into different groups, and then integrating information from different groups using 1×1 convolution. However, the spindle block extracts features at different scales independently through different network branches, which neglects the hierarchy among different scales and introduces otherwise avoidable redundancy into the network. Moreover, a naive fusion strategy like 1×1 convolution fails to emphasize informative features that could be important for effective multi-scale feature fusion.

In this paper, we propose an efficient second-order spindle block (ESAS) to tackle the aforementioned drawbacks. Specifically, an efficient multi-scale module (EMS) is devised

*Corresponding Author. This work was supported by the Natural Science Foundation of Guangdong Province(No.2020A1515010711), the Natural Science Foundation of China(Nos.61771276 and 61871258) and the Special Foundation for the Development of Strategic Emerging Industries of Shenzhen(Nos.JCYJ20170817161845824 and JCYJ20170817161056260).

978-1-7281-1331-9/20/$31.00

to exploit the hierarchical relationship among features at different scales. Considering that large-scale features can be obtained from further processing small-scale features by using convolutional layers, EMS uses the output of the small-scale feature extraction branches as additional inputs to those for large-scale features. Benefiting from the assistance of small-scale branches, large-scale branches can directly explore large-scale features without first exploring small-scale features by themselves. As a result, we manage to remove a large number of inessential model parameters from the large-scale branches and greatly improve the efficiency of the multi-scale structure. Moreover, ESAS also integrates a novel second-order attention module (SOA) to pixel-wisely capture the inter-dependencies among channels with second-order information. By examining the feature importance adaptively at the pixel level, SOA helps the network focus on more informative features. Hence, by adding SOA before the 1×1 convolution, ESAS achieves a more effective multi-scale fusion than the original spindle block, as well as less computation needed to achieve a competent extraction of rich features. An efficient second-order attention spindle network (ESASN) for lightweight SISR can be developed by stacking ESAS blocks.

In summary, the main contributions of this paper are threefold. First, we propose to remove the redundancy of spindle network by leveraging the hierarchy of multi-scale features, which results in a new module, EMS. Secondly, by modeling the inter-dependencies among channels at the pixel level, we devise another novel module, SOA, for more informative feature fusion. Last but not the least, EMS and SOA constitute ESAS, a powerful and efficient building block for lightweight SISR. On the bedrock of ESAS, we develop ESASN, an efficient second-order attention spindle network, which achieves state-of-the-art results in lightweight SISR.

2. ESASN: EFFICIENT SECOND-ORDER ATTENTION SPINDLE NETWORK

In this section, we will first present the overall architecture of ESASN in Section 2.1 before diving into details of its building blocks, ESAS in Section 2.2 and SOA in Section 2.3.

2.1. Overall Network Architecture of ESASN

The overall architecture of our proposed ESASN is shown in Fig. 1. The input and output of ESASN are denoted by I_{lr} and I_{sr}, respectively. At the beginning, a 3×3 convolutional layer is used to extract shallow feature maps G_0:

$$G_0 = F_{sf}(I_{lr}), \tag{1}$$

where F_{sf} denotes the convolution operation. The G_0 is then used for deep feature exploration by G stacked groups of local feature fusion, where each group consists of B ESAS blocks.

The flow of each local feature fusion group can be expressed as

$$\begin{aligned} G_g &= F_g(G_{g-1}) \\ &= F_{gf}([B_{g,1}, ..., B_{g,b}, ..., B_{g,B}]), \end{aligned} \tag{2}$$

where F_g denotes the g-th group's function, F_{gf} is the 1×1 convolution for local feature fusion, G_g denotes the output of the g-th group, and $B_{g,b}$ is the output of the b-th block inside the g-th group.

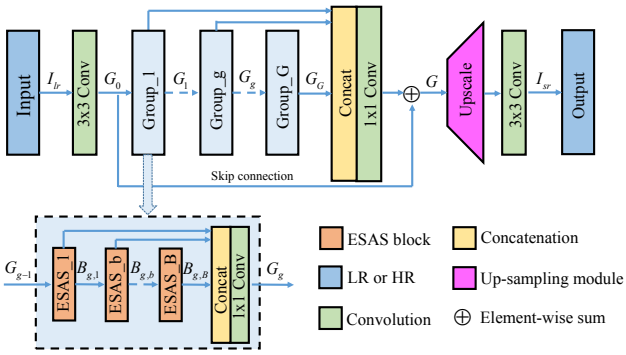


Fig. 1. The architecture of our proposed efficient second-order attention spindle network (ESASN)

The output G of the whole feature extraction is then generated by global feature fusion with skip connections:

$$G = F_{Gf}([G_1, ..., G_g, ..., G_G]) + G_0, \tag{3}$$

where F_{Gf} is a 1×1 convolution.

The final output I_{sr} is given by

$$I_{sr} = F_{last}(F_{up}(G)), \tag{4}$$

where F_{last} is a 3×3 convolution, and F_{up} denotes upsampling, which is ESPCN [13] in our paper.

2.2. ESAS: Efficient Second-order Attention Spindle Block

In this subsection, we introduce the ESAS block and its EMS module. ESAS is an advanced version of the spindle block [11], targeting at more efficient multi-scale feature extraction.

In order to produce large-scale features, a multi-scale feature extraction module usually entails large convolution kernels or many convolutional layers. For example, as shown in Fig. 2(a), the large-scale branch in the spindle block [11] generates large-scale features by stacking convolutional layers, where two 3×3 convolutional layers can achieve the same receptive field as a 5×5 convolutional layer with fewer parameters [14]. Stacking convolutional layers for large-scale branches can be regarded as first extracting small-scale features, and then using the extracted features to further explore

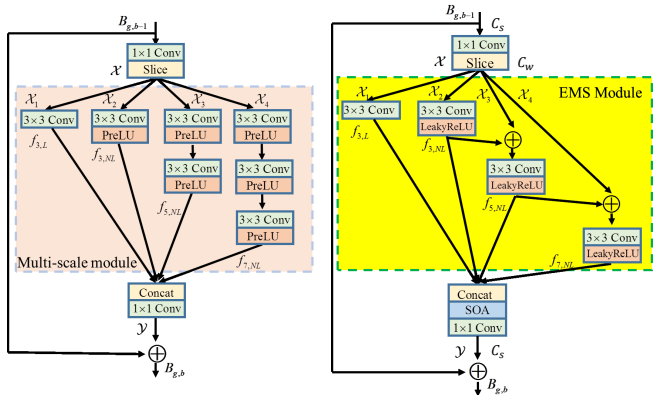
large-scale features. Inspired by this observation, we make large-scale branches directly use the features provided from small-scale branches to assist feature exploration, rather than individually extracting features from scratch.

Following this inspiration, we propose our EMS module in the ESAS block as shown in Fig. 2(b). Firstly, ESAS uses a dimension extension 1×1 convolution to produce high-dimensional feature maps $\mathcal{X} \in \mathbb{R}^{C_w \times H \times W}$. Then, the feature maps are sliced into four groups $\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3$ and $\mathcal{X}_4$ fed into four different branches for multi-scale information exploration. The first two branches in EMS are similar to the spindle block, except that we replace the $PreLU$ with $LeakyReLU$ to reduce the computation caused by the activation functions during the training phase:

$$f_{3,L} = Conv_3(\mathcal{X}_1) \tag{5}$$

$$f_{3,NL} = LeakyReLU(Conv_3(\mathcal{X}_2)), \tag{6}$$

where the subscript $_3$ means that each point of this branch's output has a 3 by 3 receptive field at the corresponding inputs, the subscript $_{L/NL}$ represents that this branch is used for linear/nonlinear information exploration, $Conv_3$ denotes a 3×3 convolution and $LeakyReLU$ is the activation function.



(a) Spindle block (b) Our ESAS block

Fig. 2. Block comparison. (a) The spindle block proposed in [11]. (b) Our efficient second-order attention spindle (ESAS) block

EMS uses the outputs of small-scale branches as additional inputs to larger-scale branches while reducing the number of convolutional layers. Element-wise addition is used to merge the raw inputs and the ones extracted by small-scale branches. The outputs of large-scale branch are then given by

$$f_{5,NL} = LeakyReLU(Conv_3(f_{3,NL} + \mathcal{X}_3)), \tag{7}$$

where $+$ denotes the element-wise addition. This merging operation connects information from the second branch to that of the third branch and provides small-scale features for the third branch to facilitate further extraction of large-scale features. Likewise, the output of the fourth branch is given by

$$f_{7,NL} = LeakyReLU(Conv_3(f_{5,NL} + \mathcal{X}_4)). \tag{8}$$

An SOA module, which will be elaborated in Section 2.3, is applied before the 1×1 feature fusion convolution to avoid ignoring nuanced features:

$$\mathcal{Y} = Conv_1(SOA(Concat(f_{3,L}, f_{3,NL}, f_{5,NL}, f_{7,NL})), \tag{9}$$

where $Conv_1$ means the 1×1 convolution, and $Concat$ denotes the concatenation operation. Then the final output of our ESAS is $B_{g,b} = \mathcal{Y} + B_{g,b-1}$.

2.3. SOA: Second-order Attention Module

In this subsection, we introduce the design of SOA module for smarter feature fusion.

Several CNN-based methods [6, 15, 16] rescaled the feature maps using the channel attention (CA) mechanism, which models the inter-dependencies among channels. However, for image restoration tasks, some detail-sensitive features are more prone to being ignored when fed into a channel descriptor with the global average pooling or maximum pooling. In order to keep as much details as possible, SOA generates an attention map by modeling the inter-dependencies among different channels adaptively at the pixel level. Inspired by the mixed high-order attention for person re-identification in [17], and taking into account that we want a lightweight network, we build our SOA module by using only second-order information, as shown in Fig. 3.

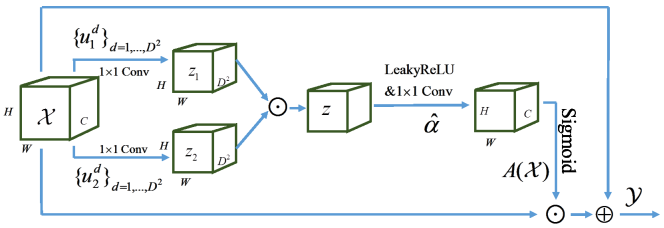


Fig. 3. Second-order attention module

In SOA, the second-order information is expressed by a linear quadratic polynomial predictor:

$$a(x) = \langle w, \otimes x \rangle \tag{10}$$

where $x \in \mathbb{R}^C$ denotes a local descriptor at a specific spatial location of $\mathcal{X}$, C denotes the number of channels, $\otimes x$ is the second-order outer-product of x, w corresponds to the weights of the elements in $\otimes x$, and $\langle ., . \rangle$ means the inner product of two same-sized tensors. According to tensor decomposition [18], we approximate w as the sum of a finite number of rank-1 tensors, $w = \sum_{d=1}^{D^2} \alpha^d u_1^d \otimes u_2^d$, where

$u_1^d \in \mathbb{R}^C, u_2^d \in \mathbb{R}^C$ are vectors, $\otimes$ denotes outer-product, α^d is the weight for d-th rank-1 tensor, and D^2 denotes a finite number. As shown in [17], we can rewrite Eq.(10) as

$$\begin{aligned} a(x) &= \langle w, \otimes x \rangle \\ &= \left\langle \sum_{d=1}^{D^2} \alpha^d u_1^d \otimes u_2^d, \otimes x \right\rangle \\ &= \sum_{d=1}^{D^2} \alpha^d \prod_{s=1}^{2} \langle u_s^d, x \rangle \\ &= \langle \alpha, z \rangle \\ &= 1^T (\alpha \odot z), \end{aligned} \tag{11}$$

where $\alpha = \left[\alpha^1, ..., \alpha^d, ..., \alpha^{D^2}\right]^T$ is the weight vector, $z = \left[z^1, ..., z^d, ..., z^{D^2}\right]^T$ with $z^d = \prod_{s=1}^{2} \langle u_s^d, x \rangle$, $\odot$ is Hadamard Product and 1^T is a row vector of ones. We introduce the auxiliary matrix P to obtain a vector-like predictor $\hat{a}(x) \in \mathbb{R}^C$, then Eq.(11) becomes

$$\hat{a}(x) = P^T(\alpha \odot z), \tag{12}$$

where $P \in \mathbb{R}^{D^2 \times C}$. Since P, α are parameters to be learned, we can integrate P, α into $\hat{\alpha} \in \mathbb{R}^{D^2 \times C}$. Then Eq.(12) can be rewritten as

$$\hat{a}(x) = \hat{\alpha}^T z. \tag{13}$$

Then, we introduce a non-linear activation function into SOA to further improve the representation capacity. Using the Sigmoid function as a proper gating function, we get the final attention map as

$$A(x) = Sigmoid(\hat{\alpha}^T \sigma(z)), \tag{14}$$

where σ denotes an arbitrary non-linear activation function, which we use $LeakyReLU$ in our experiments. $A(x) \in \mathbb{R}^C$ and the value of each entry lies in the interval $[0, 1]$. Extending from $A(x)$, defined on a local descriptor x, to $A(\mathcal{X})$, defined on $3D$ feature maps, we obtain $A(\mathcal{X}) = \left[A(x_{(1,1)}), ..., A(x_{(i,j)}), ..., A(x_{(H,W)})\right]$, where $x_{(i,j)}$ indicates a local descriptor at point (i, j) of $\mathcal{X}$. Then we get our rescaled feature maps by $\mathcal{X} \odot A(\mathcal{X})$. Inspired by the success of residual blocks, we add the input to the rescaled feature maps, the final output of SOA module is given by $\mathcal{Y} = \mathcal{X} \odot A(\mathcal{X}) + \mathcal{X}$.

3. EXPERIMENTS

3.1. Experimental Setting

We describe the experimental settings including datasets, evaluation metrics, and training details.

Datasets and metrics. We selected 800 training images from the DIV2K dataset [19] as the training set. For testing, we use five standard benchmark datasets: Set5 [20], Set14 [21], B100 [22] and Urban100 [23]. The SR results are evaluated with the signal to noise ratio (PSNR) and the structural similarity index (SSIM)[24] on the Y channel of transformed YCbCr space.

Implementation details. In our experiments of ESASN, the number of groups and blocks per group were $G = 6, B = 6$, and the channel number before and after dimension extension 1×1 convolution were $C_s = 48$ and $C_w = 96$, respectively. ESASN-S was built with $G = 4, B = 4, C_s = 36, C_w = 64$, and the depth-wise convolution was introduced to further reduce the computation. We set $D^2 = C_w//4$, and set $groups = 4$ in the 1×1 convolution of SOA module. In each training batch, we randomly cropped 16 patches with a size of 48×48 from the LR images as input. Data augmentation was performed on the training set, including data randomly rotated by $90°, 180°, 270°$ and flipped horizontally. Our model was trained with the ADAM optimizer to minimize the $L1$ loss function, where the parameters of the optimizer are $\beta_1 = 0.9, \beta_2 = 0.999$, and $\epsilon = 10^{-8}$. The leaning rate is initially set to 4×10^{-4} which decreased by half every 2×10^5 iterations of back-propagation.

3.2. Effects of EMS module and SOA module

In this subsection, we perform experiments to verify the effectiveness of the proposed the EMS and SOA modules. In order to check whether they are helpful to the efficiency of spindle block, we compare four variants of the spindle network: 1) Spindle network (SN) was built by the plain spindle block almost the same as Fig. 2(a) except that $PreLU$ was replaced by $LeakyReLU$. 2) Efficient spindle network (ESN) was built by the EMS module based spindle block. 3) Second-order attention spindle network (SASN) was built by the spindle block with the SOA module. 4) Two ESAS block based ESASNs with different numbers of model parameters were built in order to make a fair comparison: ESASN_1 was obtained by aligning the depth with other variants, and ESASN_2 was built by aligning the number of parameters.

Table 1. Effects of the SOA module and the EMS module. "✓" means the corresponding module is used while "×" means not. PSRN and SSIM are calculated on Set14(2×).

Network	Params(K)	SOA	EMS	PSNR	SSIM
SN	1300.323	×	×	33.69	0.9180
ESN	909.723	×	✓	33.66	0.9180
SASN	1343.523	✓	×	33.77	0.9190
ESASN_1	952.923	✓	✓	33.67	0.9185
ESASN_2	1331.631	✓	✓	33.79	0.9192

According to Table 1, we can observe that the EMS module reduces the number of parameters of SN by about 30% while retaining a similar performance. This indicates that our

Table 2. Comparison on benchmark datasets. Best results are **highlighted**.

Network	Scale	Params(K)	Mult-Adds(G)	Set5		Set14		B100		Urban100	
				PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
DRRN[3]	2	297	6796.9	37.74	0.9591	33.23	0.9136	32.05	0.8973	31.23	0.9188
MemNet[8]	2	677	2662.4	37.78	0.9597	33.28	0.9142	32.08	0.8978	31.31	0.9195
CARN[10]	2	1592	222.8	37.76	0.9590	33.52	0.9166	32.09	0.8978	31.92	0.9256
LFFN[11]	2	1522	342.8	37.95	0.9597	-	-	32.20	0.8994	32.39	0.9299
MSRN[7]	2	5930	1365.4	38.08	0.9607	33.70	0.9186	32.23	0.9002	32.29	0.9303
ESASN	2	1332	305.7	**38.10**	**0.9608**	**33.79**	**0.9192**	**32.25**	**0.9005**	**32.42**	**0.9312**
ESASN-S	2	173	39.8	37.82	0.9598	33.34	0.9155	32.03	0.8980	31.45	0.9215
DRRN[3]	3	297	6796.9	34.03	0.9244	29.96	0.8349	28.95	0.8004	27.53	0.8378
MemNet[8]	3	677	2662.4	34.09	0.9248	30.00	0.8350	28.96	0.8001	27.56	0.8376
CARN[10]	3	1592	118.8	34.29	0.9255	30.29	0.8407	29.06	0.8034	28.06	0.8493
LFFN[11]	3	1534	153.6	34.43	09266	-	-	29.13	0.8059	28.34	0.8558
MSRN[7]	3	6114	625.7	34.46	0.9278	30.41	0.8437	29.15	0.8064	28.33	0.8561
ESASN	3	1436	147.3	**34.50**	**0.9280**	**30.46**	**0.8445**	**29.17**	**0.8070**	**28.39**	**0.8575**
ESASN-S	3	231	24.2	34.08	0.9245	30.12	0.8381	28.96	0.8016	27.64	0.8410
DRRN[3]	4	297	6796.9	31.68	0.8888	28.21	0.7720	27.38	0.7284	25.44	0.7638
MemNet[8]	4	677	2662.4	31.74	0.8893	28.26	0.7723	27.40	0.7281	25.50	0.7630
CARN[10]	4	1592	90.9	32.13	0.8937	28.60	0.7806	27.58	0.7349	26.07	0.7837
LFFN[11]	4	1531	87.9	32.15	0.8945	-	-	27.52	0.7377	26.24	0.7902
MSRN[7]	4	6078	349.8	32.26	0.8960	28.63	0.7836	27.61	0.7380	26.22	0.7911
ESASN	4	1415	96.4	**32.26**	**0.8966**	**28.69**	**0.7844**	**27.64**	**0.7385**	**26.28**	**0.7926**
ESASN-S	4	219	21.4	31.91	0.8912	28.42	0.7775	27.44	0.7316	25.63	0.7706

EMS can reduce network redundancy while maintaining the capability of extracting multi-scale features. What's more, the SOA module improves the performance of PSNR from 33.69 dB to 33.77 dB by increasing only 3% number of parameters of SN. This indicates that the SOA module is helpful to fuse multi-scale information sufficiently at a low cost. Finally, with both the SOA module and the EMS module, our ESASN_2 achieves a better performance than SASN and enjoys the benefit of using fewer parameters.

3.3. Comparisons with the State-Of-The-Arts

We compare our ESASN with five state-of-the-art SISR methods, including DRRN [3], MemNet [8], CARN [10], LFFN [11], MSRN [7]. The results are shown in Table 2. The parameters and Muti-adds of models are also given for a intuitive comparison. Mult-adds is calculated by assuming that the spatial resolution of HR is 1280×720. The results show that our ESASN obtains the best PSNR and SSIM against all the other SISR networks, across the four test benchmarks. Especially, even though MSRN[7] has five times as many parameters as ESASN, our ESASN still performs better than MSRN. This indicates that our design of ESAS block endows the network with the capability of achieving better performance with fewer parameters.

We also provide visual comparisons of various methods in Fig. 4. It can be observed that our ESASN is able to restore clearer and more visually pleasing images, compared with other methods.

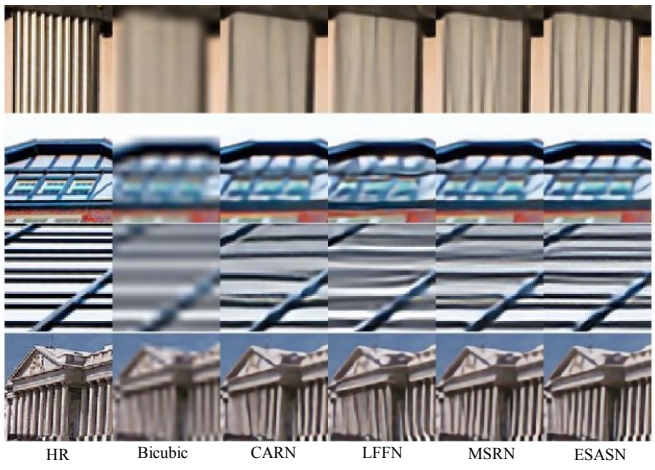


Fig. 4. Visual comparison for $\times 4$ SR on "img024", "img034", "img067", "img070" from the Urban100 Dataset.

Both quantitative and qualitative results demonstrate that our ESASN outperforms CARN and LFFN, reaching the state-of-the-art results in lightweight SISR.

4. CONCLUSION

In this paper, we propose an ESASN for lightweight single image super-resolution. Specifically, an EMS module is proposed to reduce the number of parameters while retrieving multi-scale features. Meanwhile, an SOA module is proposed to emphasize informative features adaptively at pixel-level. By combining EMS with SOA, an ESAS block can fully explore rich features for super-resolution tasks. ESASN achieves a better trade-off between efficiency and effectiveness by stacking ESAS blocks. Extensive experiments demonstrate that the proposed ESASN outperforms other lightweight SISR methods.

5. REFERENCES

[1] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang, "Learning a deep convolutional network for image super-resolution," in *ECCV*. Springer, 2014, pp. 184–199.

[2] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *CVPR*, 2016, pp. 1646–1654.

[3] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee, "Deeply-recursive convolutional network for image super-resolution," in *CVPR*, 2016, pp. 1637–1645.

[4] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *CVPR Workshops*, 2017, pp. 136–144.

[5] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu, "Residual dense network for image super-resolution," in *CVPR*, 2018, pp. 2472–2481.

[6] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu, "Image super-resolution using very deep residual channel attention networks," in *ECCV*, 2018, pp. 286–301.

[7] Juncheng Li, Faming Fang, Kangfu Mei, and Guixu Zhang, "Multi-scale residual network for image super-resolution," in *ECCV*, 2018, pp. 517–532.

[8] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu, "Memnet: A persistent memory network for image restoration," in *ICCV*, 2017, pp. 4539–4547.

[9] Wenming Yang, Xuechen Zhang, Yapeng Tian, Wei Wang, Jing-Hao Xue, and Qingmin Liao, "Deep learning for single image super-resolution: A brief review," *IEEE Transactions on Multimedia*, 2019.

[10] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn, "Fast, accurate, and lightweight super-resolution with cascading residual network," in *ECCV*, 2018, pp. 252–268.

[11] Wenming Yang, Wei Wang, Xuechen Zhang, Shuifa Sun, and Qingmin Liao, "Lightweight feature fusion network for single image super-resolution," *IEEE Signal Processing Letters*, vol. 26, no. 4, pp. 538–542, 2019.

[12] Zheng Hui, Xiumei Wang, and Xinbo Gao, "Fast and accurate single image super-resolution via information distillation network," in *CVPR*, 2018, pp. 723–731.

[13] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *CVPR*, 2016, pp. 1874–1883.

[14] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna, "Rethinking the inception architecture for computer vision," in *CVPR*, 2016, pp. 2818–2826.

[15] Yanting Hu, Jie Li, Yuanfei Huang, and Xinbo Gao, "Channel-wise and spatial feature modulation network for single image super-resolution," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[16] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang, "Second-order attention network for single image super-resolution," in *CVPR*, 2019, pp. 11065–11074.

[17] Binghui Chen, Weihong Deng, and Jiani Hu, "Mixed high-order attention network for person re-identification," in *ICCV*, 2019, pp. 371–381.

[18] Tamara G Kolda and Brett W Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[19] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang, "Ntire 2017 challenge on single image super-resolution: Methods and results," in *CVPR Workshops*, 2017, pp. 114–125.

[20] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie line Alberi Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in *BMVC*, 2012, pp. 135.1–135.10.

[21] Roman Zeyde, Michael Elad, and Matan Protter, "On single image scale-up using sparse-representations," in *International Conference on Curves and Surfaces*. Springer, 2010, pp. 711–730.

[22] David Martin, Charless Fowlkes, Doron Tal, Jitendra Malik, et al., "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *ICCV*, 2001, pp. 416–423.

[23] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja, "Single image super-resolution from transformed self-exemplars," in *CVPR*, 2015, pp. 5197–5206.

[24] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al., "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.