

EFFICIENT IMAGE SUPER RESOLUTION VIA CHANNEL DISCRIMINATIVE DEEP NEURAL NETWORK PRUNING

Zejiang Hou, Sun-Yuan Kung

Princeton University

ABSTRACT

Deep convolutional neural networks (CNN) have demonstrated superior performance in image super-resolution (SR) problem. However, CNNs are known to be heavily over-parameterized, and suffer from abundant redundancy. The growing size of CNNs may be incompatible with their deployment on mobile or embedded devices. Network pruning has benefited classification tasks by removing redundant parameters and associated computation. However, it has rarely been studied for SR, because existing methods assume the channel-wise features are of equal importance to the final reconstruction. On the contrary, we show the existence of uninformative feature-maps with no contribution to the task. In order to identify and remove such uninformative channels, we propose a new pruning criterion, *Discriminant Information*, by characterizing the dependency of the output w.r.t to the hidden-layer feature-maps. Empirically, our DI-based channel pruning algorithm is able to trim the state-of-the-art SR networks significantly (e.g. 8.7x model size compression and 3.6x CPU acceleration on SRResNet), with no quantitative or visual performance loss.

Index Terms— Efficient super-resolution, neural network pruning, channel discriminativeness, discriminant information

1. INTRODUCTION

We study the single image super-resolution (SISR) problem which entails recovering a high-resolution (HR) image from a single low-resolution (LR) one, often assumed to be a bicubic downsampled (scale x2 or x4) version of the HR counterpart. SR has been widely applied in many computer vision tasks including surveillance imaging, medical imaging, and object recognition. However, the problem of recovering high frequency details from low frequency input is ill-posed, since there exists multiple HR solutions to the LR input.

To tackle it, convolutional neural network (CNN) based methods have demonstrated significant improvements in recovering accurate and photo-realistic HR images. Since the very first 3-layer SRCNN introduced by [1], more complicated and deeper network structures [2, 3, 4, 5, 6, 7, 8] are proposed to continuously enhance the SR performance. However, the huge model size and high computational complexity hinder these CNNs' adoptions on resource-limited mobile or embedded devices. Thus, designing efficient CNNs with high perfor-

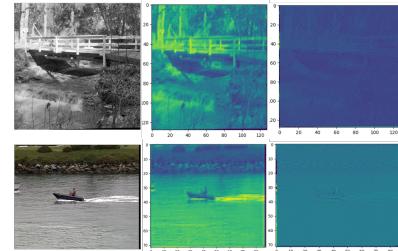


Fig. 1. Visualization of ground-truth (left), informative (middle), and uninformative feature-maps (right) of SRResNet [5].

mance becomes non-trivial. Network pruning has successfully accelerated semantic-level tasks such as object recognition and detection, by reducing computation, energy, and memory transfer costs during inference. However, little effort has been put onto SR problem, because existing CNN-based SR methods treat channel-wise features equally, assuming the removal of feature-maps would lead to unacceptable pixel-level loss.

One of our contributions is to show the existence of uninformative hidden-layer feature-maps with almost no contribution to HR reconstruction, as illustrated in Fig.1. Keeping these features would waste unnecessary computations and hinders the representation capability of the network, which motivate our exploration of a channel pruning method to directly removes the redundant feature-maps. The central of channel pruning lies in the *Local Channel Importance Ranking* (LCIR): selecting channels to remove by estimating their importance. Prior arts [9, 10, 11] adopts the "smaller filter norm less importance" criterion. However, in SISR, the output is highly correlated with the intermediate features [12]. The lack of attention to the channel-wise features' discriminativeness limits the existing pruning methods' application to SISR.

We solve the LCIR problem from the perspective of channel-wise features' discriminativeness. Specifically, we propose a new pruning criterion, i.e., *Discriminant Information* (DI), which models the statistical dependency of output on the intermediate features and removes redundant/uninformative channels with no contribution to HR reconstruction. Extensive experiments demonstrate DI-based channel pruning achieves the best performance-efficiency trade-off on state-of-the-art SR networks. Our trimmed networks perform similarly to/better than the baseline networks, but with significant model size and computational complexity savings, making them applicable to mobile devices or Internet-of-Things edge processing.

2. METHOD

2.1. Problem Settings

Given a L -layer CNN, assume the l^{th} layer has input feature-maps $\mathbf{x}_{l-1} \in \mathbb{R}^{C_{l-1} \times H_{l-1} \times W_{l-1}}$, performs convolution with filter $\mathbf{W}_l \in \mathbb{R}^{C_l \times C_{l-1} \times k \times k}$, and generates output feature-maps $\mathbf{x}_l \in \mathbb{R}^{C_l \times H_l \times W_l}$. C denotes the number of channels, H, W represents the height and width of the feature maps, and $k \times k$ is the kernel size. Such convolution has a multiplication complexity of $C_{l-1} \times C_l \times k^2 \times H_l \times W_l$. To reduce the heavy computational cost, channel pruning aims to remove the redundant feature-maps in \mathbf{x}_l in each layer to obtain a slimmer network with $C'_l < C_l$.

2.2. SR Channel Pruning Objective

Our channel pruning focus on the feature-maps' statistic. To measure the importance of channels to the final output, a criterion to model the output dependency on the feature-maps is imperative. The conditional covariance operator, proposed by Baker [13], provides a measure of the conditional dependence between random variables. Formally, let (X, Y) be a random vector with joint distribution $P_{X,Y}$. Define the cross-covariance operator associated with the pair (X, Y) as Σ_{YX} . Theorem 1 in [13] shows the existance of a unique bounded operator V_{YX} to represent the cross-covariance as:

$$\Sigma_{YX} = \Sigma_{YY}^{1/2} V_{YX} \Sigma_{XX}^{1/2} \quad (1)$$

The conditional covariance operator is then defined as:

$$\Sigma_{YY|X} = \Sigma_{YY} - \Sigma_{YY}^{1/2} V_{YX} V_{XY} \Sigma_{YY}^{1/2} \quad (2)$$

Now, we provide the following theorem due to Fukumizu et al., 2009 [14], which proves the trace of the conditional covariance operator is able to characterize the conditional dependency of the output Y onto the feature X .

Theorem 1. *For any subset \mathcal{T} of feature, i.e., $X_{\mathcal{T}}$, we have $\text{tr}\{\Sigma_{YY|X}\} \leq \text{tr}\{\Sigma_{YY|X_{\mathcal{T}}}\}$. And the equality $\text{tr}\{\Sigma_{YY|X}\} = \text{tr}\{\Sigma_{YY|X_{\mathcal{T}}}\}$ holds if and only if $Y \perp\!\!\!\perp X | X_{\mathcal{T}}$.*

For SR problem, X is the hidden-layer feature-maps, and Y is the HR output. Then, our pruning method aims to preserve the set of informative feature-maps \mathcal{T}^* which minimize $\text{tr}\{\Sigma_{YY|X_{\mathcal{T}}}\}$. In this case, the HR output becomes conditionally independent on the remaining feature-maps, removal of which causes no performance loss. Suppose we have N LR-HR training pairs $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$. Let \mathbf{x}_l^i to denote the feature-maps of l^{th} layer for the i -th LR input. Define $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_N]$, $\mathbf{X}_l = [\mathbf{x}_l^1, \dots, \mathbf{x}_l^N]$. The empirical estimate of the conditional covariance operator for l^{th} layer is given as:

$$\text{tr}\{\hat{\Sigma}_{YY|X}\} := \text{tr}\{\hat{\Sigma}_{YY} - \hat{\Sigma}_{YX}(\hat{\Sigma}_{XX} + \rho I)^{-1}\hat{\Sigma}_{XY}\} \quad (3)$$

where $\hat{\Sigma}_{YY} = \mathbf{Y}\mathbf{Y}^T$, $\hat{\Sigma}_{YX} = \mathbf{Y}\mathbf{X}_l^T$, and $\hat{\Sigma}_{XX} = \mathbf{X}_l\mathbf{X}_l^T$. Minimizing the conditional covariance operator is equivalent to maximizing the following analytical criterion, i.e., *Discriminant Information* (DI):

$$DI := \text{tr}\{(\mathbf{X}_l\mathbf{X}_l^T + \rho I)^{-1}\mathbf{X}_l\mathbf{Y}^T\mathbf{Y}\mathbf{X}_l^T\} \quad (4)$$

Algorithm 1 Layer-wise DI-based SR Channel Pruning

- 1: **Inputs:** Pretrained L -layer SR network M , training dataset $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$, pruning budget for each layer $d \in \{0.3, 0.5, 0.7\}^L$
 - 2: **Outputs:** High-performing trimmed SR network M'
 - 3: **for** $l \leftarrow 1 : L$ **do**
 - 4: Solving (5) for DI-based channel selection in layer l
 - 5: Remove $(1 - d_l)C_l$ uninformative channels from layer l
 - 6: Remove corresponding filter weights in layer $l + 1$
 - 7: Fine-tune the trimmed SR network
 - 8: Final fine-tuning of trimmed SR network
-

Formally, in each layer, DI-based SR channel pruning optimization can be established as the following channel selection:

$$\begin{aligned} \max_{\mathcal{T}} \text{tr}\{[\mathbf{X}_l\mathbf{X}_l^T + \rho I]^{-1}\mathbf{X}_l\mathbf{Y}^T\mathbf{Y}\mathbf{X}_l\} \\ \text{s.t. } |\mathcal{T}| \leq \lceil d_l C_l \rceil \end{aligned} \quad (5)$$

where d_l is a user-defined pruning budget in range $d_l \in \{0.3, 0.5, 0.7\}$, representing the fraction of preserved informative feature-maps in each layer.

2.3. Layer-wise DI-based SR Channel Pruning Algorithm

We perform channel pruning in layer-wise manner, i.e., removing uninformative channels from one layer at each step. Weights in the rest layers are inherited from previous step. Then the trimmed network is fine-tuned for one epoch, after which the pruning process moves to next layer. In each layer, channel pruning entails solving problem (5), which is NP-hard and requires combinatorial search. In practice, we propose to utilize the sequential backward elimination, a top-down approach starting with a whole set of channels and removing one channel at a time. The DI value loss after a channel is removed measures such channel's importance to HR reconstruction. The channel achieving the least DI value loss is removed. This iterative process ends when the number of preserved channels reach the user-defined pruning budget. Our layer-wise DI-based channel pruning is summarized in Algorithm 1.

3. EXPERIMENTS

3.1. Setup

To verify our method, we apply Algorithm 1 to state-of-the-art SR architectures: SRRNet/SRGAN [5], and LapSRN [4]. Following [6, 7, 8], we use the 2K resolution dataset DIV2K [15] for training the SR baselines. For testing, we evaluate our trimmed models on four benchmarks: Set5 [16], Set14 [17], BSD100 [18], and Urban100 [19]. LR images are generated using bicubic down-sampling with a scale factor of x2/x4 (a.k.a., BI degradation [5, 4]). For quantitative evaluation, we calculate PSNR and SSIM between the recovered HR images and ground-truth on Y-channel of the YCbCr color space.

3.2. Evaluation Results

To verify the effectiveness of DI-based channel pruning, we compare our trimmed models with other network compression

Table 1. Quantitative evaluation of DI channel pruning algorithm on SRResNet. Best PSNR/SSIM highlighted.

Dataset	Scale	SRResNet	B-SRResNet	DI-trim-0.3	DI-trim-0.5	DI-trim-0.7
Set5	x2	37.90/0.959	35.66/0.946	38.05/0.961	38.00/0.961	37.86/0.960
	x4	32.05/0.891	30.34/0.864	32.32/0.894	32.27/0.894	32.09/0.891
Set14	x2	33.44/0.915	31.56/0.897	33.73/0.918	33.61/0.917	33.45/0.916
	x4	28.49/0.780	27.16/0.756	28.70/0.783	28.63/0.781	28.51/0.778
BSD100	x2	32.12/0.899	-	32.20/0.900	32.09/0.900	32.04/0.898
	x4	27.58/0.735	-	27.60/0.737	27.50/0.736	27.43/0.731
Urban100	x2	31.80/0.925	28.76/0.882	32.17/0.929	31.92/0.926	31.47/0.922
	x4	25.90/0.782	24.48/0.728	26.14/0.786	25.99/0.782	25.69/0.772

Table 2. Quantitative evaluation of DI channel pruning algorithm on LapSRN. Best PSNR/SSIM highlighted.

Dataset	Scale	LapSRN	B-LapSRN	DI-trim-0.3	DI-trim-0.5	DI-trim-0.7
Set5	x2	37.25/0.957	-	37.35/0.959	37.30/0.959	37.27/0.958
	x4	31.33/0.881	30.21/0.857	31.60/0.887	31.53/0.886	31.42/0.885
Set14	x2	32.96/0.910	-	33.25/0.916	33.21/0.916	33.16/0.915
	x4	28.06/0.768	27.13/0.751	28.21/0.774	28.18/0.774	28.16/0.773
BSD100	x2	31.58/0.892	-	31.73/0.894	31.72/0.893	31.67/0.893
	x4	27.22/0.724	-	27.33/0.730	27.31/0.730	27.32/0.729
Urban100	x2	30.25/0.907	-	30.34/0.910	30.31/0.910	30.19/0.908
	x4	25.02/0.747	24.31/0.720	25.26/0.761	25.20/0.760	25.18/0.757

methods. Specifically, results from [20] are provided for comparison, where the authors proposed a network-binariization approach for efficient SR; we denote the binarized SR models as B-SRResNet and B-LapSRN. For our method, we denote our trimmed SR models as DI-trim- X , where $X \in \{0.3, 0.5, 0.7\}$ represents the per-layer pruning budget. For example, DI-trim-0.3 indicates we trim 30% channels each layer in SRResNet or LapSRN.

Quantitative results Table.1 and Table.2 show the quantitative comparisons under x2 and x4 SR in terms of PSRN and SSIM. Our DI-trimed-0.3 models performs the best on all datasets under all scales compared with both baseline networks and binarization results [20]. For example, in case of LapSRN, our DI-trim-0.3 model outperforms the baseline LapSRN by 0.27dB PSNR on Set5 under scale 4. In case of SRResNet, our DI-trim-0.3 gains 0.3dB over baseline SRResNet on Urban100 under scale 2. On the other hand, on both cases of LapSRN and SRResNet, our most efficient models (i.e. DI-trim-0.7) still outperform network-binariization [20] by a large margin (i.e. more than 1dB). Compared with [20], DI channel pruning is more effective in preserving network capacity. Moreover, DI-trim-0.7 yields around 0.1dB improvement over the baseline LapSRN for most cases. And pruning 70% channels of SRResNet causes negligible PNSR/SSIM loss.

Visual results In Fig.2, we show the visual comparison under scale 2 for the ‘baboon’ image and under scale 4 for the ‘monarch’ image. As observed, HR images generated by our DI-trim-0.7 and DI-trim-0.3 models demonstrate undetectable visual difference compared with the baseline SRResNet. However, the network binarization [20] generates more aliasing artifacts. This suggests our method has better ability to recover high-frequency details.

3.3. Model/Computation Complexity Analysis

Table.3 compares parameters, FLOPs (1280×720 p HD input), model size, and actual latency (evaluated on Set14). Binariza-

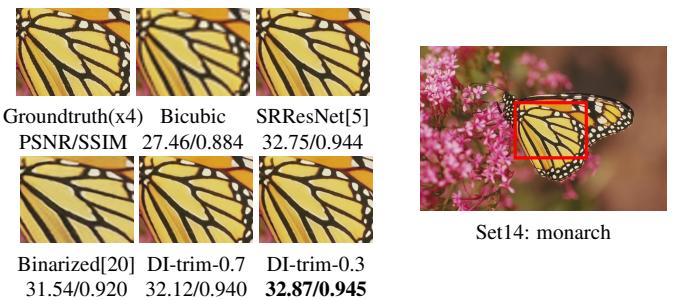
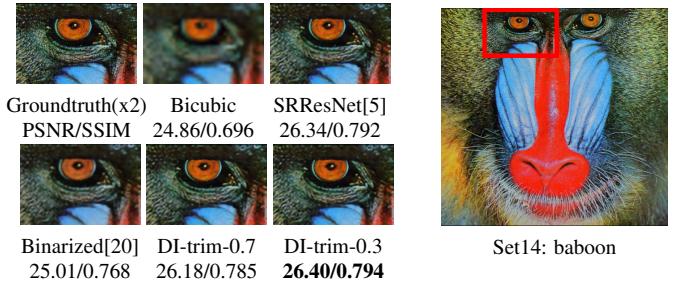


Fig. 2. Visual comparison of baseline, binarized, and trimmed SRResNet under x2/x4 scale on Set14 images.

tion [20] compresses SR model by converting floating weights into $\{-1, +1\}$, while we “shrink” the width of the network, i.e., our trimmed models are “slimmer”. For example, with pruning budget 0.7, DI channel pruning reduce the number of channels from 64 to 19, generating the most compact and fastest SR models. Under scale 2, pruning 70% channels of SRResNet achieves 8.7x size compression and 3.6x CPU acceleration, compared to 5.9x size compression for B-SRResNet. Under scale 4, pruning 70% channels of SRResNet and LapSRN achieves {9.4x compression/2.4x acceleration} and {6.8x compression/3.1x acceleration}, compared 4x and 2.3x compression for B-SRResNet and B-LapSRN, respectively. [20] only reported a *theoretical* speedup of 2x for B-SRResNet under scale 4, where our DI-trim-0.7 shows a *physical* CPU speedup of 2.4x. Furthermore, DI trimmed SR models outperform binarization by large margin in PSRN/SSIM, i.e., our method achieves the best performance-efficiency trade-off.

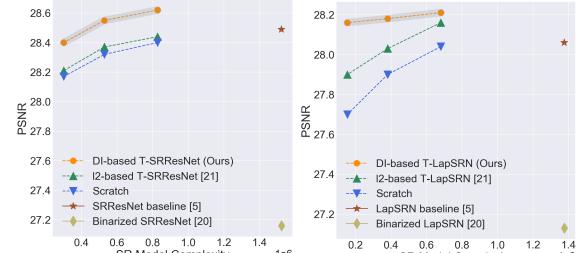


Fig. 3. Comparisons of LCIR criteria. Left: pruning results of SRResNet on Set14. Right: pruning results of LapSRN on Set14. Baseline and binarized models are also included.

3.4. Comparison of DI with other pruning method

We compare DI channel pruning with ℓ_2 -norm based pruning in CT-SRCNN [21]. We consider pruning budget in range $\{0.3, 0.5, 0.7\}$, and report testing PSNR on Set14 versus pa-

Table 3. Model complexity analysis of baseline, binarized and trimmed SR models.

Model	Scale	#Res-layers/#Filters	Weight	#Params.	FLOPs	Model Size	CPU/GPU Latency
SRResNet [5]	x2	16/64	float32	1.375M	636.9B	5.499MB	7.29s/105ms
B-SRResNet [20]	x2	16/64	{-1, +1}	1.377M	b-ops	0.928MB	-
DI-trim-0.7 (ours)	x2	16/19	float32	0.121M	74.3B	0.631MB	1.99s/89ms
DI-trim-0.3 (ours)	x2	16/44	float32	0.604M	318.4B	2.739MB	3.18s/98ms
SRResNet [5]	x4	16/64	float32	1.522M	229.6B	6.089MB	1.65s/91ms
B-SRResNet [20]	x4	16/64	{-1, +1}	1.524M	b-ops	1.518MB	-
DI-trim-0.7 (ours)	x4	16/19	float32	0.158M	88.7B	0.647MB	0.68s/84ms
DI-trim-0.3 (ours)	x4	16/44	float32	0.641M	149.3B	2.581MB	1.09s/89ms
LapSRN [4]	x4	20/64	float32	0.870M	213.7B	3.509MB	2.66s/75ms
B-LapSRN [20]	x4	20/64	{-1, +1}	0.512M	b-ops	1.494MB	-
DI-trim-0.7 (ours)	x4	20/19	float32	0.121M	24.5B	0.517MB	0.85s/66ms
DI-trim-0.3 (ours)	x4	20/44	float32	0.456M	106.3B	1.858MB	1.38s/67ms

rameters in Fig.3. Our method has superior PSNR performance against ℓ_2 -norm [21]. This is because [21] focus on the statistics of filter weights rather than feature-maps, which is insufficient for SISR. If we train the pruned network architectures from scratch, they achieve worse PSNR. This explains why we choose to compress a redundant/overly-parameterized SR networks to obtain efficient ones. Our trimmed SRResNet/LapSRN demonstrate better PSNR than baselines, even after pruning 50% channels per-layer.

3.5. Comparisons with Other Efficient SR Methods

We compare our method against other efficient SR method conducive for mobile devices, including FEQE-P [22] which is the champion of PIRM 2018 Challenge [23]. To achieve larger factor of computation saving, we replace the Subpixel upsampler [24] in SRResNet by nearest-neighbour interpolator, and apply Algorithm 1 to trim SRResNet more severely, yielding 16x size compression and 14x FLOPs reduction. As shown in Table.4, our method outperforms FEQE-P in all aspects: higher PSNR/SSIM (0.3dB gain on Set5), lower parameters and FLOPs. Compared with SRCNN [1] and VDSR [3], our method demonstrates better PSNR/SSIM with much less parameters than VDSR. SRCNN has the least amount of parameters, but far more FLOPs: SRCNN 128G vs. ours 17G, measured on 1280×720 p HD input.

Table 4. Comparisons with other efficient SR methods.

Model	Set5	Set14	BSD100	Urban100	Params.
SRCNN [1]	30.47/0.861	27.57/0.753	26.89/0.711	24.51/0.723	69K
CT-SRCNN [21]	31.48/0.884	28.18/0.767	27.28/0.725	-	110K
VDSR [2]	31.53/0.884	28.42/0.783	27.29/0.726	25.18/0.753	668K
FEQE-P [22]	31.53/0.882	28.21/0.771	27.32/0.727	25.32/0.758	96K
Ours	31.84/0.889	28.38/0.775	27.40/0.730	25.51/0.765	92K

Furthermore, Fig.4 shows the visual comparison under scale 4. Our method generates more plausible HR images with much sharper edges/textures and less artifacts than FEQE-P.

4. SUPER-RESOLVING REAL-WORLD IMAGES

Considering the constraints on communication bandwidth and power for data uploading/offloading, real-world images are subject to JPEG compression [25]. Super-resolving JPEG-compressed LR images is more challenging, due to the extra artifacts introduced by JPEG. To address it, we adopt data-augmentation by combining BI-degraded LR images with the JPEG-compressed LR images for finetuning our efficient SR

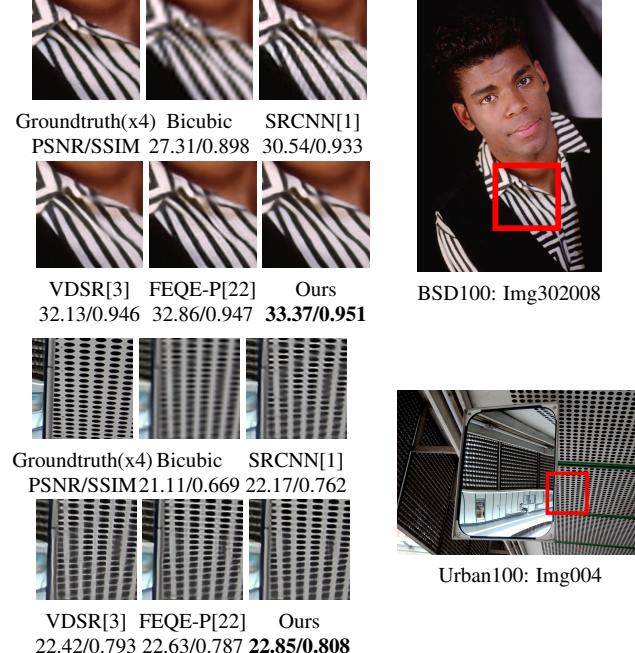


Fig. 4. Visual comparison with other efficient SR methods under x4 scale on BSD100/Urban100 images.

models. Table.5 reports the performance on JPEG-compressed Set14. Combining JPEG and down-sampling, our total image compression becomes 27x. Compared to scale-2 baseline SRResNet, our DI-trim-0.3 model achieves 0.1dB PSNR gain with and 2x complexity reduction.

Table 5. Super-resolving JPEG compressed LR images.

Models	Scale	JPEG	Image \downarrow	PSNR/SSIM	#Params.
SRResNet	x2	Q75	27x	30.81/0.808	1.522M
DI-trim-0.3	x2	Q75	27x	30.75/0.847	0.158M
DI-trim-0.5	x2	Q75	27x	30.88/0.850	0.363M
DI-trim-0.7	x2	Q75	27x	30.94/0.852	0.641M

5. CONCLUSION

We introduce a *Discriminant Information* based CNN channel pruning algorithm for efficient super-resolution. Inspired by conditional covariance characterizing the input-output conditional dependency, we propose the DI criterion to estimate the feature-maps' importance to the HR image reconstruction. Quantitative and qualitative experiments on state-of-the-art SR networks demonstrate our method generates comparable/better results than existing efficient SR methods, and achieve significant reduction in model size and computation complexity.

6. REFERENCES

- [1] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang, “Image super-resolution using deep convolutional networks,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2015.
- [2] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee, “Accurate image super-resolution using very deep convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1646–1654.
- [3] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee, “Deeply-recursive convolutional network for image super-resolution,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1637–1645.
- [4] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang, “Deep laplacian pyramid networks for fast and accurate super-resolution,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 624–632.
- [5] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al., “Photo-realistic single image super-resolution using a generative adversarial network,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.
- [6] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee, “Enhanced deep residual networks for single image super-resolution,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 136–144.
- [7] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu, “Residual dense network for image super-resolution,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2472–2481.
- [8] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu, “Image super-resolution using very deep residual channel attention networks,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 286–301.
- [9] Song Han, Jeff Pool, John Tran, and William Dally, “Learning both weights and connections for efficient neural network,” in *Advances in neural information processing systems*, 2015, pp. 1135–1143.
- [10] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf, “Pruning filters for efficient convnets,” *arXiv preprint arXiv:1608.08710*, 2016.
- [11] Yang He, Guoliang Kang, Xuanyi Dong, Yanwei Fu, and Yi Yang, “Soft filter pruning for accelerating deep convolutional neural networks,” *arXiv preprint arXiv:1808.06866*, 2018.
- [12] Juncheng Li, Faming Fang, Kangfu Mei, and Guixu Zhang, “Multi-scale residual network for image super-resolution,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 517–532.
- [13] Charles R Baker, “Joint measures and cross-covariance operators,” *Transactions of the American Mathematical Society*, vol. 186, pp. 273–289, 1973.
- [14] Kenji Fukumizu, Francis R Bach, Michael I Jordan, et al., “Kernel dimension reduction in regression,” *The Annals of Statistics*, vol. 37, no. 4, pp. 1871–1905, 2009.
- [15] Eirikur Agustsson and Radu Timofte, “Ntire 2017 challenge on single image super-resolution: Dataset and study,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 126–135.
- [16] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel, “Low-complexity single-image super-resolution based on nonnegative neighbor embedding,” 2012.
- [17] 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.
- [18] 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,” *Iccv Vancouver*, 2001.
- [19] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja, “Single image super-resolution from transformed self-exemplars,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5197–5206.
- [20] Yinglan Ma, Hongyu Xiong, Zhe Hu, and Lizhuang Ma, “Efficient super resolution using binarized neural network,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.
- [21] Haoyu Ren, Mostafa El-Khamy, and Jungwon Lee, “Ct-srcnn: cascade trained and trimmed deep convolutional neural networks for image super resolution,” in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 1423–1431.
- [22] Thang Vu, Cao Van Nguyen, Trung X Pham, Tung M Luu, and Chang D Yoo, “Fast and efficient image quality enhancement via desubpixel convolutional neural networks,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 0–0.
- [23] Yochai Blau, Roey Mechrez, Radu Timofte, Tomer Michaeli, and Lihi Zelnik-Manor, “The 2018 pirm challenge on perceptual image super-resolution,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 0–0.
- [24] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1874–1883.
- [25] William B Pennebaker and Joan L Mitchell, *JPEG: Still image data compression standard*, Springer Science & Business Media, 1992.