IMAGE SUPER-RESOLUTION VIA DEEP AGGREGATION NETWORK

Xinya Wang[1], *Jiayi Ma*[1,*], *Junjun Jiang*[2]

[1]Electronic Information School, Wuhan University, Wuhan 430072, China
[2]School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China

ABSTRACT

Deep convolutional neural networks (CNNs) have recently made a considerable achievement in the single-image super-resolution (SISR) problem. Most CNN architectures for SISR incorporate skip connections to integrate features, and treat them equally. However, this neglects the discrimination of features, and consequently, achieving relatively poor performance. To address this problem, we introduce a deep aggregation network that merging extraction and aggregation nodes in a tree structure, which can aggregate features progressively. In particular, we rescale the information in the aggregation node by modelling the interaction between channels, which shares the same insight on the attention mechanism for improving the discriminative ability of network. In the extraction node, we introduce an mlpconv layer into a dense unit that is parallel to the convolutional layer and can improve the nonlinear mapping capability, where the residual learning is utilized to accelerate the training process. Extensive experiments conducted on several publicly available datasets have demonstrated the superiority of our model over state-of-the-art in objective metrics and visual impressions.

Index Terms— Super-resolution, convolutional neural network, aggregation, mlpconv layer, attention mechanism

1. INTRODUCTION

Single-image super-resolution (SISR) aiming to reconstruct a high-resolution (HR) image from a single low-resolution (LR) input image, has been widely studied in computer vision, medical imaging, satellite imaging, *etc*. SISR is essentially to recover high-frequency details from low-frequency data, and it poses an ill-posed and challenging problem due to the inevitable loss of information in the image degeneration process. To solve this problem, various methods have been investigated, including interpolation-, reconstruction-, and learning-based methods [1, 2, 3]. Possessing the strong nonlinear expressiveness, convolutional neural network (CNN) based methods have become increasingly popular in recent years for solving the aforementioned ill-posed problem [4, 5].

To learn a nonlinear LR-HR mapping, Dong *et al*. [6] first introduced CNN into SISR in an end-to-end manner, called SRCNN, which has shown its superiority to non-deep learning methods even if it relies on just a small receptive field. The SRCNN is subsequently improved by incorporating long or short and multi-path skip connections, such as VDSR [7], DRCN [8] and DRRN [9]. In general, a deeper network can obtain more nonlinearity and larger receptive fields, and significant efforts have been made to achieve this goal, such as EDSR [10], MemNet [11] and RDN [12]. However, as the network growing deeper, the features extracted by convolutional layers would be hierarchical. How to make full use of such features to recover more details is still an open problem.

In the recent past, to better propagate information, researchers have been focusing on designing a network connecting the features densely. DCSCN [13] firstly introduced the densely connection in extraction network by only one dense block, and subsequently skip connections between several dense blocks in SRDenseNet [14] were utilized, performing better reconstruction results. Furthermore, different levels of information extracted by dense block were concated for construction in RDN [12]. Nevertheless, the existing scenarios typically involve two major disadvantages. On the one hand, it is difficult for each layer to abstract ample nonlinear features via a single filter. On the other hand, when the densely connected features at different levels are employed for the next operation or for reconstruction, they are treated equally, neglecting the discrimination of the information.

In this paper, we proposed a deep aggregation network named SRDAN to address the aforementioned drawbacks. In particular, we unify the aggregation and extraction nodes in a tree structure to learn richer combinations from the feature hierarchy. Similar to the HDA structure [15], our SRDAN iteratively merges the hierarchical features through an aggregation module, as shown in Fig. 1. Rather than concating the features at different levels directly, we propagate the output of an aggregation node through all the previous states and progressively aggregate and deepen the representations. In the aggregation node, channel-wise features are combined and compressed for efficiency. Before sending features to the next stage, this node would learn to select features of relative importance by attention mechanism, so that the network is discriminative and simultaneously offers a more efficient

*Corresponding author (e-mail: jyma2010@gmail.com).

This work was supported by the National Natural Science Foundation of China under Grant no. 61773295.

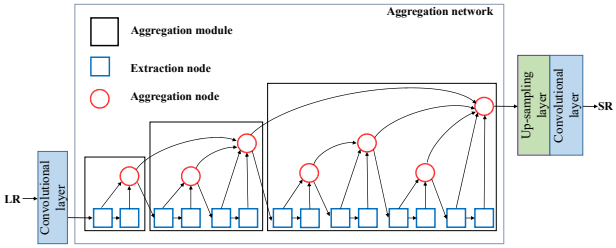

Fig. 1. The whole structure of the proposed SRDAN.

use of computation resource. In the extraction node, we modify the dense block and introduce the multilayer perceptron convolutional (mlpconv) layer [16] to improve the nonlinear mapping capability. Parallel to the convolutional layer, they both have access to the additional inputs from all previous dense units and pass on information that should be preserved. Subsequently, local residual learning is utilized to adaptively preserve the local feature after concating the state of previous dense units. Our method is evaluated on standard benchmark data sets, which outperforms state-of-the-art approaches in terms of peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM).

Contributions. Our main contributions include the following three aspects. First, we propose a new deep aggregation network for SISR, which is able to outperform the state-of-the-art. Second, an mlpconv layer is introduced into a dense unit and combined with local residual learning, which can avoid information loss after numerous layers and improve network capability to learn nonlinear mapping from LR and HR patches. Third, we introduce the attention mechanism for feature selection, enhancing the discriminative learning ability and decreasing the parameters of the network.

2. PROPOSED METHOD

2.1. Overall Architecture

As shown in Fig. 1, our SRDAN consists of three parts: low-level feature extraction, aggregation network and reconstruction. The output is I^{SR} when a low-resolution image I^{LR} is taken as the initial input. First, we use a single convolution layer to extract a shadow feature F_0

$$F_0 = H_{3\times3}(I^{\mathrm{LR}}), \tag{1}$$

where $H_{3\times3}$ denotes the convolutional operation with kernel size 3×3. Subsequently, the first output F_0 is applied to the aggregation network, as the rich feature F_{RF} is obtained as

$$F_{\mathrm{RF}} = A(F_0), \tag{2}$$

where A denotes the operation of the deep aggregation network, which contains aggregation modules, merging aggregation nodes and extraction nodes. In the reconstruction network, we upscale the feature into high-resolution space by

$$F_{\mathrm{UF}} = U(F_{\mathrm{RF}}), \tag{3}$$

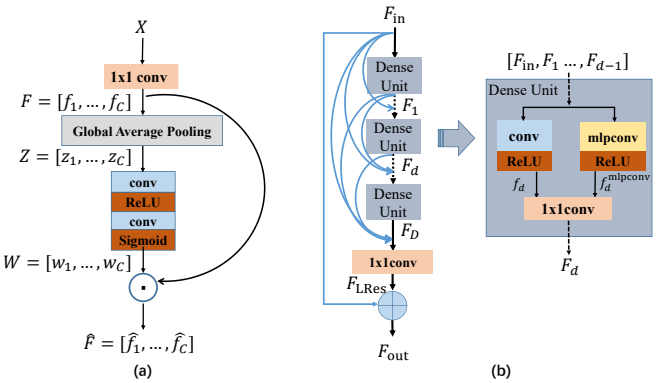

Fig. 2. Structures of (a) aggregation and (b) extraction nodes.

where U conducts the upscale, similar to that in EDSR, and F_{UF} is the upscaled feature. Finally, the SR result is reconstructed by a convolutional layer

$$I^{\mathrm{SR}} = H_{3\times3}(F_{\mathrm{UF}}). \tag{4}$$

Our method is optimized with ℓ_1 loss function. Given the training set $\{(I_i^{\mathrm{LR}}, I_i^{\mathrm{HR}})\}_{i=1}^{N}$, which consists of N LR patches and corresponding HR ones. Thereby, the goal of training SRDAN is to minimize the ℓ_1 loss between the SR results and their corresponding HR counterparts

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\left\|I_i^{\mathrm{SR}} - I_i^{\mathrm{HR}}\right\|_1. \tag{5}$$

2.2. Aggregation Network

Due to the fact that hierarchical feature combinations are benefit for reconstruction when the network is growing deeper [12], we introduce a hierarchical deep aggregation tree as the aggregation network, as shown in Fig. 1. Specifically, in each aggregation module, the extraction node and aggregation node are assembled in a tree-structured way to aggregate different levels of representation. To improve the depth of the aggregation module, the output of the aggregation node is sent to the backbone to merge deeper feature iteratively.

2.2.1. Aggregation Node

Sharing the same insight of the attention mechanism, the aggregation node aims to merge representations from different levels and select informative features simultaneously, as shown in Fig. 2a. At the beginning, a convolutional layer with a 1×1 kernel is utilized for integrating spatial information. Given an input X, the output is obtained by

$$F = H_{1\times1}(X). \tag{6}$$

Due to the fact that low-frequency information extracted in LR space are abundant even trivial, taking all concated features for information abstraction would consume too much

resource. Therefore, we adopt the attention mechanism to select useful features. To learn the inter dependencies between channels, global spatial information is collected into a channel descriptor for expressing the whole image by global average pooling. Let $F = [f_1, \ldots, f_C]$ be the input, which has C feature maps with size of $H \times W$. The channel-wise statistic $Z = [z_1, \ldots, z_C] \in R^C$ is then generated by shrinking F through the spatial dimension $H \times W$

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} f_c(i, j), \quad c = 1, \ldots, C, \tag{7}$$

where $f_c(i, j)$ is the value at the position (i, j) of the c-th feature f_c. To make use of statistics for capturing channel-wise dependencies, the nonlinear and non-mutually-exclusive interactions should be learned adaptively to ensure the discrimination on multiple channels [17]. Consequently, we employ two convolutional layers followed by a sigmoid function to acquire the final channel-wise weight

$$W = s(H_{3\times3}(\sigma(H_{3\times3}(Z)))), \tag{8}$$

where s and σ represent the sigmoid and ReLU activation, respectively. Finally, the feature F is reweighed along the c-th channel to perform feature selection through emphasising informative features and suppressing less useful ones

$$\hat{F} = F \cdot W. \tag{9}$$

2.2.2. *Extraction Node*

In the extraction node, we propose an improved dense block to abstract high-dimensional information through a set of dense units. The features of these units will be concated and the final output is obtained via local residual learning, as shown in Fig. 2b. Given F_{in} as input of the extraction node with D dense units, for the d-th dense unit, the input is a composite of all the outputs from the previous layers and the original input, which can be formulated as $[F_{\text{in}}, F_1, \ldots, F_{d-1}]$. Thus the output can be expressed as

$$F_d = \mathcal{D}([F_{\text{in}}, F_1, \ldots, F_{d-1}]), \tag{10}$$

where $\mathcal{D}$ indicates the operation in a single dense unit.

The details of $\mathcal{D}$ is shown in the right part of Fig. 2b. We extract deep local features by using a convolutional layer paralleled with an mlpconv layer, and output G feature maps as follows:

$$f_d = H_{3\times3}([F_{\text{in}}, F_1, \ldots, F_{d-1}]), \tag{11}$$

$$f_d^{\text{mlpconv}} = H^{\text{mlpconv}}([F_{\text{in}}, F_1, \ldots, F_{d-1}]), \tag{12}$$

where the superscript mlpconv denotes the operation of the mlpconv layer. To control the depth of the output in one dense unit, we use a convolutional layer with a kernel size of 1×1 to integrate the aforementioned features:

$$F_d = H_{1\times1}([f_d, f_d^{\text{mlpconv}}]), \tag{13}$$

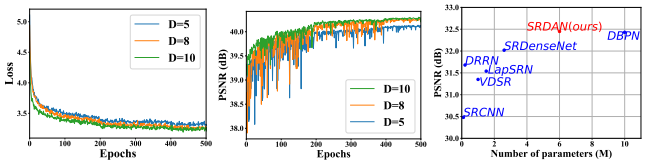

Fig. 3. Loss (left) and PSNR (middle) of our aggregation network at different D. The right plot reports the parameters and PSNR of different methods evaluated on Set5 at factor 4.

in which feature F_d has G feature maps.

All the outputs of the dense units and input F_{in} are concated as $[F_{\text{in}}, F_1, \ldots, F_D]$ after extracting several high-level features using the same kind of dense unit. For residual learning, the local residual should maintain the same size as the identity map, and hence we adopt a 1×1 layer as follows:

$$F_{\text{LRes}} = H_{1\times1}([F_{\text{in}}, F_1, \ldots, F_D]), \tag{14}$$

where F_{LRes} denotes local residual.

Finally, the output of the extraction node is obtained by

$$F_{\text{out}} = F_{\text{LRes}} + F_{\text{in}}. \tag{15}$$

2.3. Implementation Settings

In the aggregation node, the output map number is cut in half of the input, *i.e.*, X in Eq. (6) has $2C$ feature maps. In the extraction node, each convolutional layer and mlpconv layer are followed by an ReLU except for the 1×1 convolutional layer. We set the number of dense units as 10, *i.e.* $D = 10$. Convolutional layers in dense unit have $G = 16$ filters. The first and the last convolutional layers have 64 and 3 filters, as we output color images. Zero padding is used to keep the same size before the upsacaling layer. For the upscaling layer, we follow ESPCNN [5] to upscale the coarse resolution features to the expected ones. We use 800 training images from the DIV2K dataset [18] as the training set. In each training batch, 16 LR color patches of size 96×96 are extracted as inputs. Our model is trained by the ADAM optimizor [19] with default settings. The initial leaning rate is set to 10^{-4} and then decreases to half every 2×10^5 iterations of back-propagation. We use PyTorch to implement our model with a GTX 1080Ti.

3. EXPERIMENTS

3.1. Model Analysis

The effective way to deepen our aggregation network is to increase the number of the dense units in the extraction node. Therefore, we investigate the influence of the hyperparameter D, including the loss and PSNR during training process, as in the left two plots of Fig. 3. From the results, we see that as D increases from 8 to 10, the convergence of the loss function and PSNR is quite similar, and $D = 10$ performs only slightly better. Therefore, we set $D = 10$ as the default value. There

Table 2. Public benchmark test results (PSNR/SSIM) for scale factor x2, x3, x4. Red: the best; blue: the second best.

Dataset	Scale	Bicubic	SRCNN [6]	VDSR [7]	DRRN [9]	LapSRN [20]	SRDenseNet [14]	DBPN [21]	Ours
Set5	x2	33.65/0.9299	36.66/0.9542	37.53/0.9590	37.74/0.9591	37.52/0.9591	—	38.09/0.9600	38.12/0.9609
	x3	30.39/0.8682	32.75/0.9090	33.67/0.9210	34.03/0.9244	33.82/0.9227	—	—	34.55/0.9280
	x4	28.42/0.8104	30.48/0.8626	31.35/0.8830	31.68/0.8888	31.54/0.8855	31.58/0.8853	32.43/ 0.8971	32.45/0.8972
Set14	x2	30.24/0.8688	32.45/0.9067	33.05/0.9130	33.23/0.9136	33.08/0.9130	—	33.85/0.9190	33.83/0.9195
	x3	27.55/0.7742	29.30/0.8215	29.78/0.8320	29.96/0.8349	29.79/0.8320	—	—	30.44/0.8448
	x4	26.00/0.7027	27.50/0.7531	28.02/0.7680	28.21/0.7721	28.19/0.7720	28.36/0.7701	28.75/0.7861	28.80/0.7863
BSD100	x2	29.56/0.8431	31.36/0.8879	31.90/0.8960	32.05/0.8973	31.80/0.8950	—	32.27/0.9000	32.26/0.9005
	x3	27.21/0.7385	28.41/0.7863	28.83/0.7990	28.95/0.8004	28.82/0.7973	—	—	29.18/0.8069
	x4	25.96/0.6675	26.90/0.7101	27.29/0.7260	27.38/0.7284	27.32/0.7280	27.38/0.7310	27.67/0.7393	27.69/0.7395
Urban100	x2	26.88/0.8403	29.50/0.8946	30.77/0.9140	31.23/0.9188	30.41/0.9101	—	32.56/0.9310	32.58/0.9320
	x3	24.46/0.7349	26.24/0.7969	27.14/0.8290	27.53/0.8378	27.07/0.8272	—	—	28.53/0.8596
	x4	23.14/0.6577	24.52/0.7221	25.18/0.7540	25.44/0.7638	25.21/0.7553	26.05/0.7819	26.38/0.7950	26.42/0.7960

Table 1. Performance on different choices of extraction node evaluated on the DIV2K validation dataset. Red: the best.

	Method in [14]	Method in [12]	Ours
PSNR (dB)	39.635	39.781	39.928

are several alternatives to the extraction node. To validate the effectiveness of our improved dense block, we conduct experiments on different choices, such as the dense block in [14] and residual dense block in [12], all of which have 10 layers with the growth rate of 32 to keep the parameters at the same level. The results are reported in Table 1. After iterating 8×10^5 back-propagation, our improved dense block obtains the highest PSNR, benefited from the improvement of non-linear ability when introducing the mlpconv layer.

3.2. Comparisons with State-of-The-Art Methods

We compare our SRDAN with 6 state-of-the-art SR methods, including SRCNN [6], VDSR [7], DRRN [9], LapSRN [20], SRDenseNet [14] and DBPN [21]. Four publicly available datasets are used for evaluation such as Set5 [22], Set14 [23], BSD100 [24] and Urban100 [25]. The results are reported in Table 2. The two widely used metrics PSNR and SSIM are evaluated on the SR results at the upscaling factors 2, 3 and 4. The blank in the table means that the corresponding algorithm does not conduct at that factor in the original paper. Due to the benefit of the progressive aggregation, our SRDAN can achieve the best performance. Although the PSNR values of our SRDAN are only slightly better than the latest results, DBPN, our model has much less parameters, as shown in the right plot of Fig. 3. Therefore, our SRDAN is able to reach a better trade-off between the model size and the performance.

In Fig. 4, we display visual comparisons on three images from BSD100. It can be seen that our method performs better

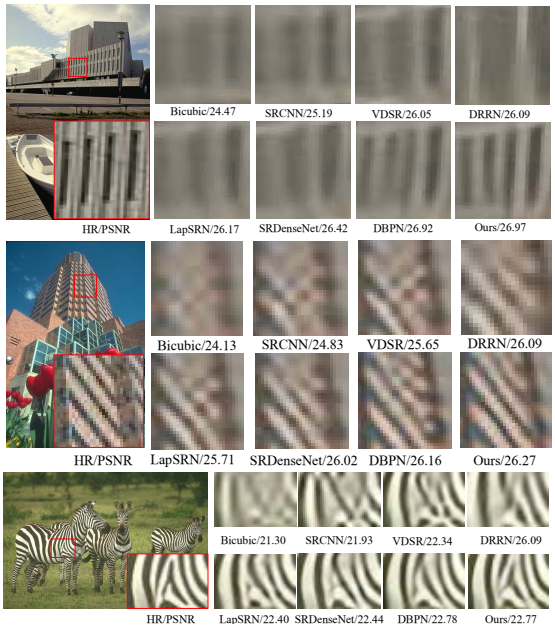

Fig. 4. Qualitative comparison of super resolution results for "img78004" (top), "img86000" (middle), and "img253027" (bottom) from BSD100 with an upscaling factor of 4.

on the structure of the objects, while other methods generate results with noticeable artifacts.

4. CONCLUSION

In this study, we propose a novel deep aggregation network called SRDAN for SISR, which can achieve the state-of-the-art performance in terms of PSNR and SSIM. A key characteristic of our SRDAN is to integrate the extraction node and the aggregation node into a tree. This enables us to interactively abstract the features and selectively preserve the useful ones, which not only improves the non-linear and discriminable ability, but also decreases the model size.

5. REFERENCES

[1] Jing Tian and Kai-Kuang Ma, "A survey on super-resolution imaging," *Signal, Image and Video Processing*, vol. 5, no. 3, pp. 329–342, 2011.

[2] Junjun Jiang, Xiang Ma, Chen Chen, Tao Lu, Zhongyuan Wang, and Jiayi Ma, "Single image super-resolution via locally regularized anchored neighborhood regression and nonlocal means," *IEEE Transactions on Multimedia*, vol. 19, no. 1, pp. 15–26, 2017.

[3] Junjun Jiang, Yi Yu, Zheng Wang, Suhua Tang, and Jiayi Ma, "Ensemble super-resolution with a reference dataset," *IEEE Transactions on Cybernetics*, 2019.

[4] Chao Dong, Chen Change Loy, and Xiaoou Tang, "Accelerating the super-resolution convolutional neural network," in *ECCV*, 2016, pp. 391–407.

[5] 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.

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

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

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

[9] Ying Tai, Jian Yang, and Xiaoming Liu, "Image super-resolution via deep recursive residual network," in *CVPR*, 2017, pp. 2790–2798.

[10] 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.

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

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

[13] Yamanaka Jin, Shigesumi Kuwashima, and Takio Kurita, "Fast and accurate image super resolution by deep cnn with skip connection and network in network," in *ICNIP*, 2017, pp. 217–225.

[14] Tong Tong, Gen Li, Xiejie Liu, and Qinquan Gao, "Image super-resolution using dense skip connections," in *ICCV*, 2017, pp. 4809–4817.

[15] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell, "Deep layer aggregation," *arXiv preprint arXiv:1707.06484*, 2017.

[16] Min Lin, Qiang Chen, and Shuicheng Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

[17] Jie Hu, Li Shen, and Gang Sun, "Squeeze-and-excitation networks," *arXiv preprint arXiv:1709.01507*, 2017.

[18] Eirikur Agustsson and Radu Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *CVPR Workshops*, 2017, pp. 126–135.

[19] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[20] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang, "Deep laplacian pyramid networks for fast and accurate superresolution," in *CVPR*, 2017, pp. 624–632.

[21] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita, "Deep backprojection networks for super-resolution," in *CVPR*, 2018, pp. 1664–1673.

[22] 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.

[23] Roman Zeyde, Michael Elad, and Matan Protter, "On single image scale-up using sparse-representations," in *ICCS*, 2010, pp. 711–730.

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

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