



Improved differentiable neural architecture search for single image super-resolution

Yu Weng¹ · Zehua Chen¹ · Tianbao Zhou¹

Received: 26 May 2020 / Accepted: 9 December 2020 / Published online: 14 January 2021
© The Author(s) 2021

Abstract

Deep learning has shown prominent superiority over other machine learning algorithms in Single Image Super-Resolution (SISR). In order to reduce the efforts and resources cost on manually designing deep architecture, we use differentiable neural architecture search (DARTS) on SISR. Since neural architecture search was originally used for classification tasks, our experiments show that direct usage of DARTS on super-resolutions tasks will give rise to many skip connections in the search architecture, which results in the poor performance of final architecture. Thus, it is necessary for DARTS to have made some improvements for the application in the field of SISR. According to characteristics of SISR, we remove redundant operations and redesign some operations in the cell to achieve an improved DARTS. Then we use the improved DARTS to search convolution cells as a nonlinear mapping part of super-resolution network. The new super-resolution architecture shows its effectiveness on benchmark datasets and DIV2K dataset.

Keywords SISR · DARTS · Convolution cell · Nonlinear mapping part

1 Introduction

As a branch of computer vision [20], image super-resolution technique reconstructs a higher-resolution image from the observed lower-resolution counterpart, which is known as a notoriously challenging ill-posed inverse procedure. Benefit from the strong capacity of extracting effective high-level abstractions which bridge the low-resolution (LR) space and high-resolution (HR) space, many SISR methods [19, 21, 22, 33] based on deep neural network have applied in this field recently.

Designing a good neural network architecture is time-consuming and laborious, in order to reduce the efforts and resources cost on manually designing network architecture, scholars and researchers put their attention on neural architecture search (NAS). Neural architecture search [24, 31, 45] is currently a popular method for searching for an effective architecture, which has been applied to image classification [44], image segmentation [39], object recognition [45] and other fields. The key problem of SISR methods based on deep neural network is to search a deep neural network architecture with better performance for SR.

Generally speaking, NAS contains reinforcement learning (RL) [44, 45], evolutionary algorithm (EA) [31], gradient descent (GD) [4, 39], Bayesian optimization [29] and so on. Compared with most NAS methods based on mentioned above, DARTS as a prominent representative of GD-based methods, greatly reduces the training time and hardware resources [25]. This leads us to apply DARTS to search for a cell-based architecture instead of the nonlinear mapping part in super-resolution network. However, in original DARTS, with the increase of training rounds, the entire proxy network tends to choose more skip-connect operations in searched architecture, which leads to the collapse of the model, and it is demonstrated by our experiments that this problem is even more serious when we apply DARTS directly to super-resolution tasks. Therefore, it is necessary for DARTS

This article belongs to the Topical Collection: *Special Issue on P2P Computing for Deep Learning*

Guest Editors: Ying Li, R.K. Shyamasundar, Yuyu Yin, Mohammad S. Obaidat

✉ Zehua Chen
18810723551@163.com

Yu Weng
dr_wengyu@126.com

Tianbao Zhou
tianbaochou@163.com

¹ Minzu University of China, 27 Zhongguancun South Avenue, Beijing, China

to have made some improvements for the application in the field of SR. In this paper, we optimize the connection between the intermediate nodes and the output node of the cell. Because SISR task requires huge memory and is difficult to overfit, we remove pooling operations to accelerate training procedure. The entire proxy network prefers to choose many skip connection operations in searched architecture, so we add identity mapping in convolutions operations to avoid aforementioned phenomenon.

2 Related works

Single image super-resolution refers to the task of restoring high resolution images from one low-resolution observations of the same scene. At present, Image super-resolution research can be divided into three main categories: interpolation-based, reconstruction-based and learning-based methods. The early methods are interpolation-based methods, such as bicubic interpolation [18] and Lanczos resampling [11], are very fast and simple but it also has some obvious shortcomings. Firstly, it assumes that the change in the gray value of a pixel is a continuous and smooth process, but in fact this assumption is not completely true. Secondly, during the reconstruction process, SR images are calculated based on only a pre-defined conversion function, and image degradation model is not taken into account, which often results in blurred, jagged and other phenomena in the restored images. Considering the upper problems, reconstruction-based SR methods [8, 27, 35, 40] start from the degradation model of the image. It is assumed that the high-resolution image undergoes proper motion transformation, blurring, and noise to obtain a low-resolution image. This method restricts the generation of super-resolution images by extracting key information from low-resolution images and combining prior knowledge of unknown super-resolution images. It can generate flexible and sharp details, but also brings some problems. With the scale factor increases, the performance of many reconstruction-based methods degrades rapidly, and these methods are usually time-consuming.

With the development of machine learning, machine learning is widely used in different fields, such as intelligent transportation systems [14], recommendation [13, 41], data translation [23], extubation failure [6] and dynamic reconfiguration [12]. Besides, deep learning is widely used in super-resolution reconstruction algorithms. Learning-based method [5, 9, 10, 19, 21, 22, 32–34] uses a large amount of training data to learn a certain correspondence between the low-resolution image and the high-resolution image, and then predicts the high-resolution image corresponding to the low-resolution image based on the learned mapping relationship. Learning-based method

has a better performance, but the design of neural network requires a lot of manpower, computing resources and time.

In order to reduce the efforts and resources cost on manually designing architecture, neural architecture search has attracted the attention of researchers. Most NAS approaches can be categorized in two modalities: macro search and micro search. Macro search algorithms aim to directly discover the entire neural network, reinforcement learning (RL) [44, 45], evolutionary algorithm (EA) [31] and Bayesian optimization [29] are the representatives, but these methods need long training time and high resource consumption. Micro search algorithms aim to discover neural cells and design a neural architecture by stacking many copies of the discovered cell. Since NASNet [45] successfully search neural cells on NASNet search space, more researchers propose their methods [4, 26, 43] based on NASNet search space. Notably, DARTS is simpler than many existing approaches as it does not involve any controllers [1, 30, 44], hypernetworks [3] or performance predictors [24]. Besides, DARTS reduces structure search time to several GPU days, its simpler than many existing approaches. Considering that the current application of deep learning in super-resolution is mainly to learning the mapping relationship between low-resolution images and corresponding high-resolution images, using DARTS to search a nonlinear mapping network is a very worthwhile thing to try. In summary, considering training time and resource consumption, we use a simpler neural architecture search method (DARTS) to find an efficient architecture for super-resolution tasks.

3 Original DARTS for SR tasks

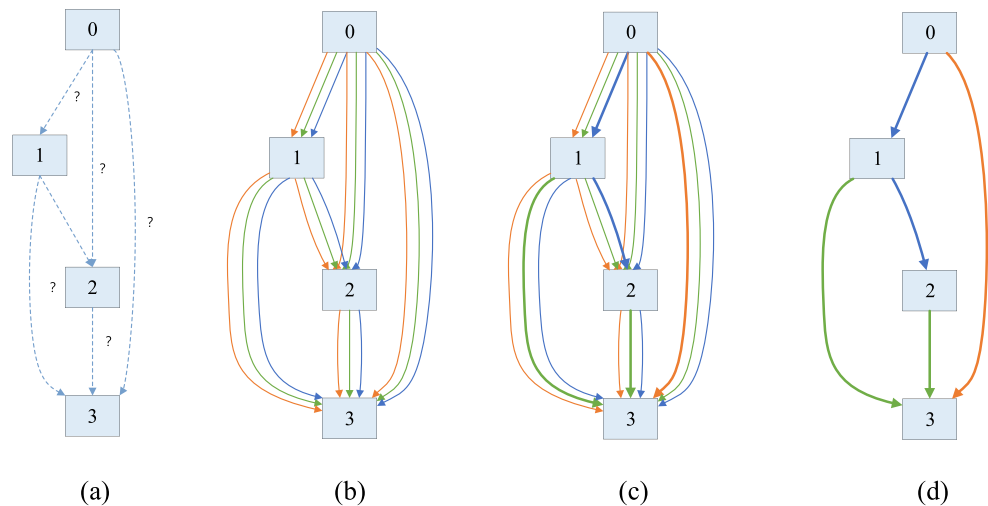
3.1 Preliminary of differentiable architecture search

For the case of convolution neural networks, DARTS [25] searches for a normal cell and a reduction cell to build up the final architecture. A cell is a directed acyclic graph constructed by N nodes. Each node x^i is a feature map in cell, and each edge (i, j) is associated with some operation $O^{(i,j)}$, which are used to change x^i . For convolution cells, each cell has two inputs and one output. Two inputs are the outputs of previous two cells. The output of the cell is obtained by applying a reduction operation (e.g. concatenation) to all intermediate nodes in the cell. Each intermediate node is computed based on all of its predecessors, the details are shown in Fig. 1:

$$x^i = \sum_{j < i} O^{(j,i)} x^j \quad (1)$$

where $O^{(j,i)}$ means an operation on x^j , summing all the obtained feature maps to get x^i . In DARTS, the author specifies that feature map of each intermediate node is

Fig. 1 An overview of DARTS: **a** Operations on the edges are initially unknown. **b** Continuous relaxation of the search space by placing a mixture of candidate operations on each edge. **c** Joint optimization of the mixing probabilities and the network weights by solving a bilevel optimization problem. **d** Inducing the final architecture from the learned mixing probabilities. This figure quoted from DARTS



obtained by operating the feature map of all previous nodes. Therefore, the task of learning the cell is transformed into learning operations on its edges.

Suppose O is a collection of all operations (e.g., convolution, zero, skip-connection) where each operation represents some function o to be applied to x^i . The blending weights for node i and node j are represented by the vector $\alpha^{(i,j)}$. To make the search space continuous, the categorical choice of a particular operation is relaxed as a softmax over all possible operations:

$$\bar{o} = \sum_{o \in O} \frac{\exp(\alpha_o^{(i,j)})}{\sum_{o' \in O} \exp(\alpha_{o'}^{(i,j)})} o(x) \quad (2)$$

Where $\alpha_o^{(i,j)}$ and $\alpha_{o'}^{(i,j)}$ mean an operation's weight on edge (i, j) , $o(x)$ means an operation to be applied to feature map x . The weight represents the importance of the operation. The larger the weight value, the more important the operations is.

In the following, the author refers to α as the encoding of the architecture and the get the corresponding operations based on the learned blending weight α . For the search procedure, we denote L_{train} and L_{val} as the training and validation loss respectively. Then the architecture parameters are learned with the following bilevel optimization problem:

$$\min L_{val}(w^*(\alpha), \alpha) \quad (3)$$

$$s.t. \quad w^*(\alpha) = \arg \min_w L_{train}(w, \alpha) \quad (4)$$

In DARTS, the author proposed an approximate iterative optimization procedure where w and α are optimized by alternating between gradient descent steps in the weight and architecture space respectively. The details are shown in Table 1.

As shown in Table 1, in DARTS, the architecture α and the weight w of the neural network are optimized by alternate iteration. In the gradient back propagation phrase, the neural network weights w are updated with the loss of the training set, and the architecture α is updated with the validation loss. In step k , given the current architecture α_{k-1} , the proxy network obtains w_k by moving w_{k-1} in the direction of minimizing the training loss $L_{train}(w_{k-1}, \alpha_{k-1})$. Then, keeping the weights w_k fixed and updating the architecture so as to minimize the validation loss after a single step of gradient descent w.r.t the weights:

$$L_{val}(w_k - \epsilon \nabla_w L_{train}(w_k, \alpha_{k-1}), \alpha_{k-1}) \quad (5)$$

where ϵ is the learning rate for this virtual step.

After get the continuous architecture α , we retained 2 strongest predecessors for each intermediate node, where the strength of an edge is defined as:

$$\max_{o \in O, o \neq zero} \frac{\exp(\alpha_o^{(i,j)})}{\sum_{o' \in O} \exp(\alpha_{o'}^{(i,j)})} \quad (6)$$

Table 1 The algorithm of bilevel optimization in DARTS

Algorithm 1 darts-differentiable architecture search.

Create a mixed operation $\bar{o}^{(i,j)}$ parameterized by $\alpha^{(i,j)}$ for each edge (i, j)

While not converged do

1. Update weight w by descending $\nabla_w L_{train}(w, \alpha)$

2. Update architecture α by descend $L_{val}(w_k - \epsilon \nabla_w L_{train}(w_k, \alpha_{k-1}), \alpha_{k-1})$

Replace $\bar{o}^{(i,j)} = \arg \max_{o \in O} \alpha_o^{(i,j)}$ for each edge (i, j)

Then we replace every mixed operation as the most likely operation by taking the argmax. More technical details can be found in original DARTS paper.

3.2 The phenomenon of performance drop caused by intractable skip connections

In PDARTS [7], a severe issue underlying DARTS has been found. Namely, after a certain searching epoch, the number of skip connections increases dramatically in the selected architecture, which results in poor performance of the selected architecture. why it happens? From another perspective, the authors relaxed the categorical choice of a particular operation as a softmax over all possible operations in DARTS. This weighted sum resembles a basic residual model in ResNet [15, 16], which states that the identity mapping ensures that information is directly propagated back to any shallower layers. In other words, it is helpful to train a deeper neural network. In ResNet experiments, they showed that the learned residual functions in general have small responses, suggesting that identity mapping provide reasonable preconditioning. Therefore, the skip connection's corresponding architectural weight increases much faster than other operations' architectural weights. The identity mapping part and residual function part in residual module work together to reach a better result, but the final architecture is obtained by picking only the top-performing ones among all operations which break this cooperation.

In image classification task, it has been observed in DARTS that lots of skip connections are involved in the selected architecture, which make the architecture shallow and the performance poor. To see if there is the same problem in super-resolution tasks, we run DARTS two times with different random seed. In super-resolution DIV2K dataset, the alpha value of skip connection become very large when the number of search epochs is large, and

thus the number of skip-connect increases in the selected architecture as shown in the blue line in Fig. 2. Following DARTS, we select 8 top-performing operations per cell. The number of dominate skip connection operations (highest softmax (α) among all operations in that edge) occupies a large part of all operation searched by DARTS. In addition, in left picture of Fig. 2, all top-performing operations in selected architecture are skip connections, it indicates that the proxy network has not learned useful operations. Considering the above situation, we need to prevent search for too many skip-connect operations in search stage.

4 Improved DARTS for super-resolution tasks

In this chapter, we adjust the search space based on the characteristics of some notable super-resolution networks. After that, we describe our proposed SR network architecture based on differentiable architecture search. We ultimately present the proposed network and compare this network with the baseline.

4.1 Remove redundant operations

Like the search space proposed in DARTS, we search for a basic cell structure and then stack it to construct the final convolutional neural network, which is used as a nonlinear mapping part of the super-resolution network. In some super-resolution networks [21, 22], a large initial channel means that convolution operation can get more feature maps, thereby saving more information about LR images and getting higher quality SR images. Reduction cell will increase the number of channels to be output, if the initial channel number if too large, the usage of reduction cell will result in out of memory. In addition, the length and width of the feature output will be halved, which will result in information loss and more upsampling

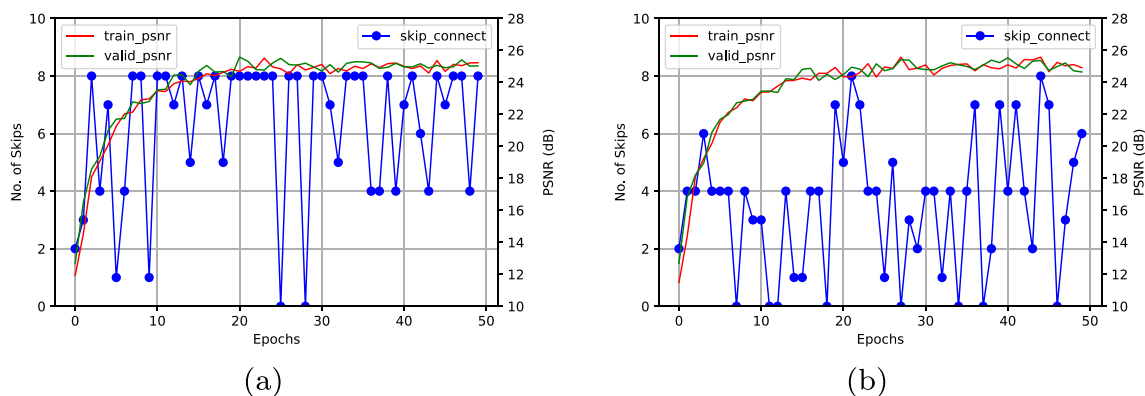


Fig. 2 The number of skip connections accounts for a large portion of all operation searched by DARTS. Train and validation PSNR are also drawn to show the level of convergence

operations. Therefore, in order to ensure that the initial channel is relatively large, we removed the reduction cell in experiment.

For SISR tasks, since the input LR image and the output HR image are strongly correlated, and the LR image is down-sampling by the HR image. Since pooling operations often leads to the information loss, so they may harm the final performance. Of course, removing redundant operations speeds up the search architecture stage. To reduce the size of parameter in cells, unlike the output of cell in DARTS, the output of cell in our search space is the mean of feature maps of all intermediate nodes in the cell.

4.2 Adding identity mapping on convolution operations

Since the proxy network tends to choose skip connection as dominant operation in all operations, we decided to add residual block unit in search space. Instead of adding extra residual module in search space, we add identity mapping in original convolution operations in DARTS. Besides, by stacking residual block unit, some notable networks [21, 22] have achieved good performance on SR tasks, it indicates that residual module is useful. Like residual block, we add skip connection to convolution neural network (The details are shown in Fig. 3). In image classification task, the architecture will prefer to search many skip-connection operations that cause the collapse of the final model. Adding skip connection in convolution operations not only improves the performance, but also avoid choosing more skip-connection in architecture search stage. Then we use

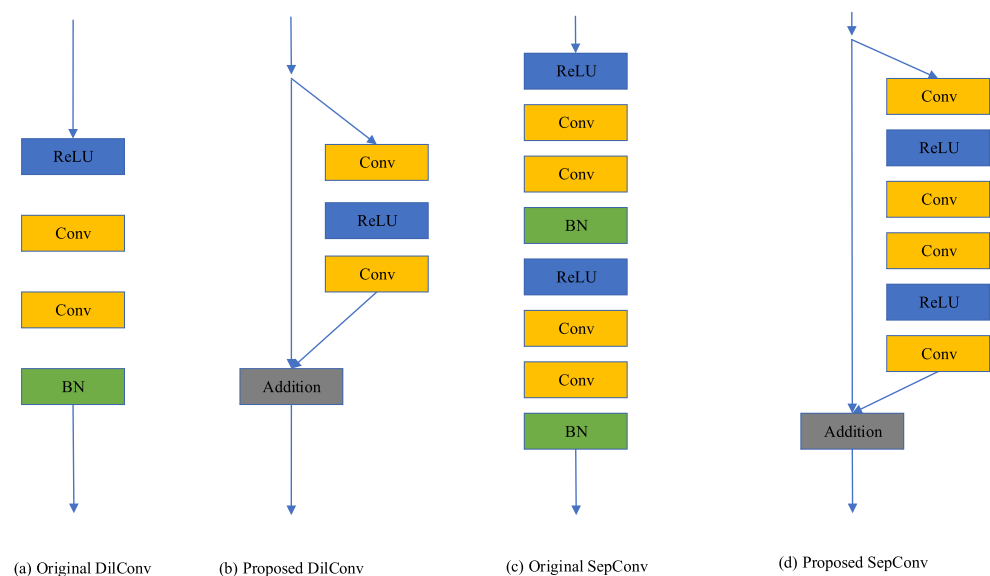
experiments to verify that this method can really bring about an improvement in the effect.

For comparison, like we did before, we run two times with different random seed, the number of skip connections keep steady in Fig. 4. In next section, we use experiment to verify that searched architecture performance better. Compared with result got by Original DARTS in Fig. 3, the result says that choosing more skip connections gets higher PSNR value during the search phase. In architecture search phase, an edge consists of six competitive operations, the entire proxy network has a better performance. Whereas, the final architecture is obtained by picking only the top-performing ones among all operations, which result in the performance drop of final architecture. In the experiment part, we verifies that the results of Improved DARTS are better than Original DARTS.

4.3 The architecture of the proposed network

At the begin of the neural network, we use convolution operations extract low-level features. Then we use cell structure that can be searched by DARTS to serve as a nonlinear mapping part of super-resolution network (Fig. 5). The network searched by DARTS also has skip-connection, so it can make the network deeper. After that, we use element-wise addition to fuse the low-level features and the high-level features got by nonlinear mapping part of super-resolution network. Finally, we use subpixel upsampling to get corresponding HR images. During the forward propagation of the entire network, the dimensions of the features remain the same.

Fig. 3 Comparison of convolution operations in original DARTS and ours



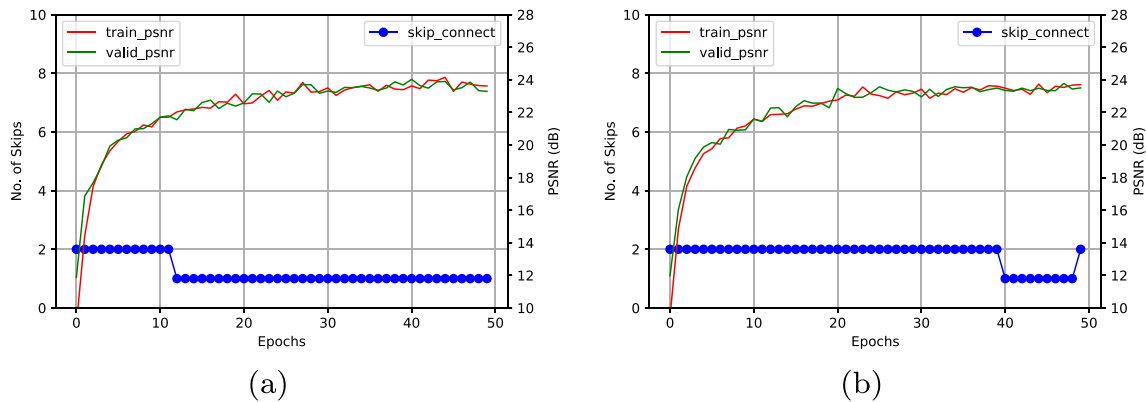


Fig. 4 The number of skip connections keeps steady as training epochs increase. Train and validation PSNR are also drawn to show the level of convergence

5 Experiments and results

The experiment includes two parts: architecture search and architecture evaluation. In the first part we use DARTS to search a cell structure (including the connection method and the operation used for the connection). In architecture evaluation phase, we train the searched model from scratch in the training process. For testing, we use five standard benchmark datasets. The SR results are evaluated with PSNR and SSIM [38] on Y channel of transformed YCbCr space.

5.1 PSNR/SSIM for measuring reconstruction quality

PSNR and SSIM are quantitative criteria in most super-resolution papers. Given two images X and Y both with N pixels, peak signal-to-noise ratio is defined as:

$$PSNR = 10 * \log_{10} \frac{MAX_I^2}{MSE} = 20 * \frac{MAX_I}{\sqrt{MSE}} \quad (7)$$

Where MSE is the mean square error between the original image and the SR image, MAX_I indicates the maximum value of the image color (8-bit sample point is represented as 255). PSNR value evaluates the quality of the image by comparing the gray value difference of the corresponding pixels of two images. The higher the PSNR value, the better the image obtained by super-resolution Structural similarity

index (SSIM) is defined as:

$$SSIM(X, Y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{XY} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (8)$$

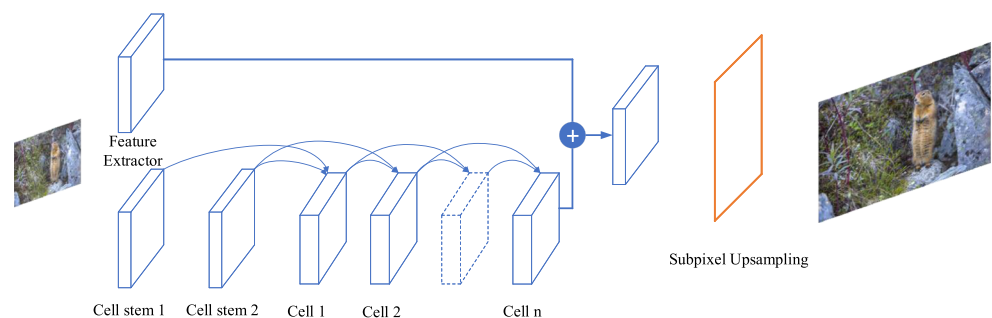
where μ_X and μ_Y represent the mean values of the images X and Y , and σ_X and σ_Y represent the standard deviations of the images X and Y . σ_{XY} represents the covariance of the images X and Y , C_1 and C_2 are constant. SSIM evaluates the similarity of the two images from three aspects: brightness, contrast, and structure. SSIM is a number between 0 and 1. The larger the SSIM value, the better the super-resolution effect.

5.2 Architecture search

We include the following operations in O : 3×3 and 5×5 proposed separable convolutions, 3×3 and 5×5 proposed dilated separable convolutions, skip-connection and zero. All operations are of stride one (if applicable) and the convolved feature maps are padded to preserve their spatial resolution. We use Conv-ReLU-Conv order for proposed separable convolution and dilated separable convolution.

Our convolutional cell consists of $N = 7$ nodes, and the output of the cell is the mean of the feature maps of all intermediate nodes in the cell (input nodes excluded). The rest of the setup follows DARTS [25]. We took a half of the training data in DIV2K dataset as validation set and

Fig. 5 Neural architecture of super-resolution



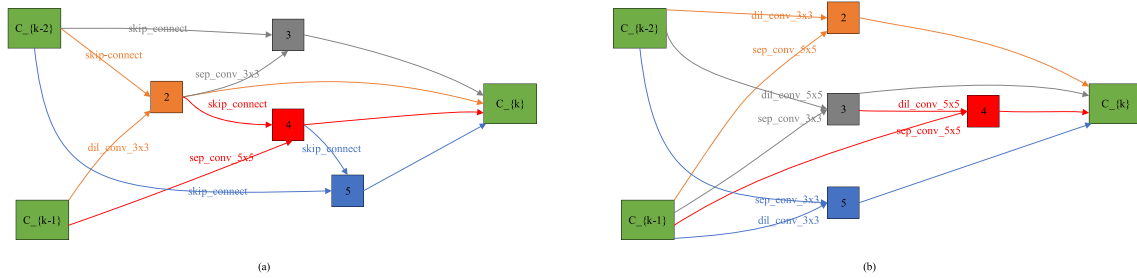
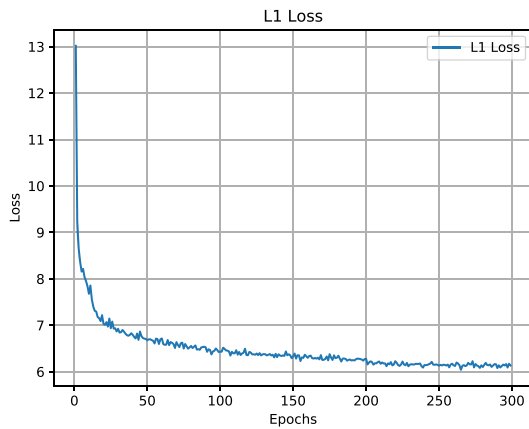


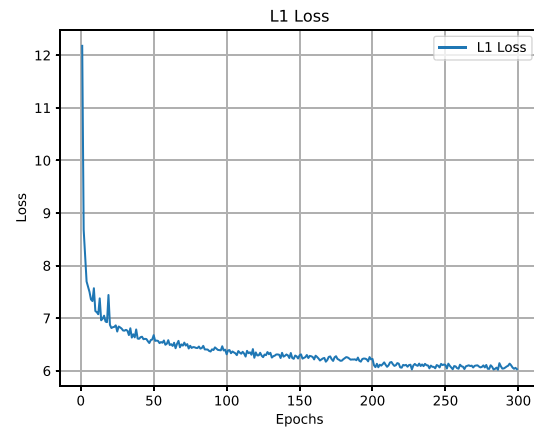
Fig. 6 a The searched result from Original DARTS. b The searched result from Improved DARTS

the other used was used as training data. In order to find a suitable architecture faster, we chose a small network with only 8 cells at this stage. The network is trained using DARTS for 50 epochs, with batch size 16 and the initial number of channels 64. We use SGD optimizer to optimize the weight w of the network, with momentum 0.9, initial

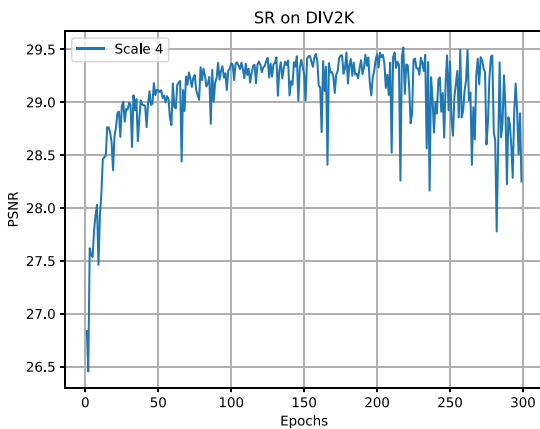
learning rate $1e - 2$ and the regularization $1e - 3$. At the same time, we use the Adam optimizer to adjust architecture α , with initial learning rate $3e - 4$, the momentum $\beta = (0.9, 0.999)$, and the regularization $1e - 3$. It took 8 hours on one NVIDIA TITAN Xp GPU. The searched results are shown in Fig. 6.



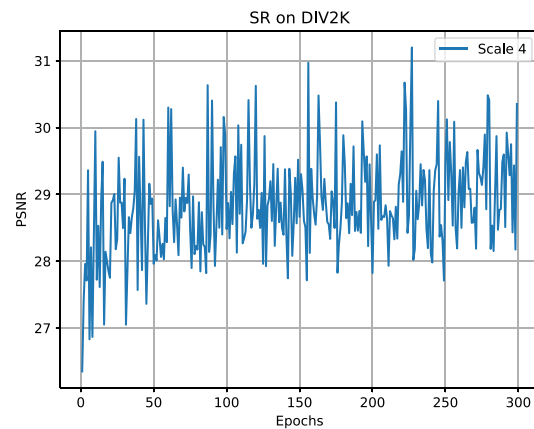
(a)



(b)



(c)



(d)

Fig. 7 The two pictures (a) and (c) represent the loss and PSNR(dB) values of the original DARTS trained on the DIV2K dataset. The two pictures (b) and (d) represent the loss and PSNR(dB) values of the improved DARTS trained on the DIV2K dataset

Table 2 Public benchmark test results (PSNR (dB))

Dataset	Bicubic	A+	SRCNN	VDSR	SRResNet	Original DARTS	Ours
Set5	28.42/0.8104	30.28/0.8603	30.48/0.8628	31.35/0.8838	32.05/0.8910	32.21/0.8949	32.21/0.8958
Set14	26.00/0.7027	27.32/0.7491	27.49/0.7503	28.01/0.7674	28.53/0.7804	28.66/0.7823	28.61/0.7832
B100	25.96/0.6675	26.82/0.7087	26.90/0.7101	27.29/0.7251	27.57/0.7354	27.51/0.7323	27.61/0.7384
Urban100	23.14/0.6577	24.32/0.7183	24.52/0.7221	25.18/0.7524	26.07/0.7839	25.96/0.7687	26.26/0.7932

Bold indicates the best performance

5.3 Architecture evaluation

5.3.1 Evaluation on DIV2K dataset

DIV2K [37] is a large collected dataset of RGB images with a large diversity of contents. The DIV2K is divided into 800 training image, 100 validation images, and 100 test images. As the test dataset ground truth is not released, we report and compare the performance on the validation dataset. We train our model with 800 training images and use 10 validation images in the training process.

We used a 20-layer large network and trained for 300 epochs, with input patch size 192, and initial channel 256 and batch size 16. Initial learning rate is set to $1e-4$ and is dropped by half for every 200 epochs. We augment the training data with random horizontal flips and 90 rotations. We pre-process all the images by subtracting the mean RGB value of DIV2K dataset. We train our model with ADAM optimizer by setting $\beta = (0.9, 0.999)$, and $\epsilon = 1e-8$. Because super-resolution model is hard to converge, so we removed the regularization part and dropout layer. It took 5 days on three NVIDIA TITAN Xp GPUs. The result is shown in Fig. 7. Compared with our baseline model EDSR, the authors use 32 residual blocks and 256 filters to get the best performance, and the model parameters of their model is 43M. While in our proposed model, the parameters of our model is 26.68M. For training, we use the RGB input patches of size 48×48 from low-resolution image with the corresponding high-resolution patches. Under the same input condition as above, the flops of EDSR is 1853 GFLOPS and the flops of our proposed model is 1162 GFLOPS. Other models do not clearly give model parameters and flops, so we don't compare with them here.

5.3.2 Benchmark results

We provide quantitative and qualitative comparisons. Besides DIV2K validation set, we evaluate our proposed model on four standard benchmark datasets: Set5 [2], Set14 [42], B100 [28], and Urban100 [17]. We compare our model with some notable methods including SRCNN [9], A+ [36], VDSR [19] and SRResNet [21]. We only compare our

model with others on scale 4. For comparison, we measure PSNR on the y channel. In Table 2, we provide a summary of quantitative evaluation on several datasets. Our model shows significant improvement than other models. Except for PSNR, we also introduce SSIM as another parameter to measure the performance of the model on benchmark datasets. On more complex datasets B100 and Urban100, improved DARTS is better than original DARTS.

6 Conclusion

In this work, we have presented a novel cell-based super-resolution method using very deep networks. We redesign convolution operations including dilation convolution and separable convolution by adding skip-connection. By removing unnecessary operations from search operation set, we shortened the search time of the architecture. Our proposed SISR model surpasses some notable works. Moreover, neural architecture search offers a feasible way for engineers to compress existing popular human-designed models.

Acknowledgements Thanks to Guosheng Yang for his guidance on this article. At the same time, I would like to thank the students in the laboratory for their suggestions and comments on this article.

References

- Baker B, Gupta O, Naik N, Raskar R (2017) Designing neural network architectures using reinforcement learning. In: 5th international conference on learning representations, ICLR 2017, Toulon, France, April 24–26, 2017. Conference Track Proceedings. <https://openreview.net/forum?id=S1c2cvqee>
- Bevilacqua M, Roumy A, Guillemot C, Alberi-Morel M (2012) Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In: British machine vision conference, BMVC 2012, Surrey, UK, September 3–7, 2012, pp 1–10. <https://doi.org/10.5244/C.26.135>
- Brock A, Lim T, Ritchie JM, Weston N (2018) SMASH: one-shot model architecture search through hypernetworks. In: 6th international conference on learning representations, ICLR 2018, Vancouver, BC, Canada, April 30 – May 3, 2018. Conference Track Proceedings. <https://openreview.net/forum?id=rydeCEhs->

4. Cai H, Zhu L, Han S (2019) ProxylessNAS: direct neural architecture search on target task and hardware. In: 7th international conference on learning representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. <https://openreview.net/forum?id=HyIVB3AqYm>
5. Chen J, Ying H, Liu X, Gu J, Feng R, Chen T, Gao H, Wu J (2020) A transfer learning based super-resolution microscopy for biopsy slice images: the joint methods perspective. *IEEE/ACM Trans Comput Biol Bioinf*
6. Chen T, Xu J, Ying H, Chen X, Feng R, Fang X, Gao H, Wu J (2019) Prediction of extubation failure for intensive care unit patients using light gradient boosting machine. *IEEE Access* 7:150960–150968
7. Chen X, Xie L, Wu J, Tian Q (2019) Progressive Differentiable Architecture Search: Bridging the Depth Gap Between Search and Evaluation. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. arXiv:1904.12760
8. Dai S, Han M, Xu W, Wu Y, Gong Y, Katsaggelos AK (2009) SoftCuts: a soft edge smoothness prior for color image super-resolution. *IEEE Trans Image Process* 18(5):969–981. <https://doi.org/10.1109/TIP.2009.2012908>
9. Dong C, Loy CC, He K, Tang X (2014) 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*, pp 184–199. https://doi.org/10.1007/978-3-319-10593-2_13
10. Dong C, Loy CC, Tang X (2016) 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*, pp 391–407. https://doi.org/10.1007/978-3-319-46475-6_25
11. Duchon CE (1979) Lanczos filtering in one and two dimensions. *J Appl Meteorol* 18(8):1016–1022
12. Gao H, Huang W, Duan Y (2020) The cloud-edge based dynamic reconfiguration to service workflow for mobile ecommerce environments: a QoS prediction perspective. *ACM Trans Internet Technol*
13. Gao H, Kuang L, Yin Y, Guo B, Dou K (2020) Mining consuming behaviors with temporal evolution for personalized recommendation in mobile marketing apps. *Proc ACM/Springer Mobile Netw Appl(MONET)*
14. Gao H, Liu C, Li Y, Yang X (2020) V2VR: Reliable Hybrid-Network-Oriented V2V Data Transmission and Routing Considering RSUs and Connectivity Probability. *IEEE Trans Intell Trans Sys*: 1–14. <https://doi.org/10.1109/TITS.2020.2983835>
15. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE conference on computer vision and pattern recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, pp 770–778. <https://doi.org/10.1109/CVPR.2016.90>
16. He K, Zhang X, Ren S, Sun J (2016) Identity mappings in deep residual networks. In: *Computer vision - ECCV 2016 - 14th European conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, pp 630–645. https://doi.org/10.1007/978-3-319-46493-0_38
17. Huang J, Singh A, Ahuja N (2015) 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*, pp 5197–5206. <https://doi.org/10.1109/CVPR.2015.7299156>
18. Keys R (1981) Cubic convolution interpolation for digital image processing. *IEEE Trans Acous Speech Sig Process* 29(6):1153–1160
19. Kim J, Kwon Lee J, Mu Lee K (2016) Accurate image super-resolution using very deep convolutional networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 1646–1654
20. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional networks. In: *Advances in neural information processing systems*, pp 1097–1105
21. Ledig C, Theis L, Huszár F, Caballero J, Cunningham A, Acosta A, Aitken A, Tejani A, Totz J, Wang Z et al (2017) Photo-realistic single image super-resolution using a generative adversarial network. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 4681–4690
22. Lim B, Son S, Kim H, Nah S, Mu Lee K (2017) Enhanced deep residual networks for single image super-resolution. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp 136–144
23. Lin B, Deng S, Gao H, Yin J (2020) A multi-scale activity transition network for data translation in EEG signals decoding. *IEEE/ACM Trans Comput Biol Bioinf*
24. Liu C, Zoph B, Neumann M, Shlens J, Hua W, Li LJ, Fei-Fei L, Yuille A, Huang J, Murphy K (2018) Progressive neural architecture search. In: *Proceedings of the European conference on computer vision (ECCV)*, pp 19–34
25. Liu H, Simonyan K, Yang Y (2018) Darts: Differentiable architecture search. arXiv:1806.09055
26. Luo R, Tian F, Qin T, Chen E, Liu T (2018) Neural architecture optimization. In: *Advances in neural information processing systems 31: annual conference on neural information processing systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pp 7827–7838. <http://papers.nips.cc/paper/8007-neural-architecture-optimization>
27. Marquina A, Osher SJ (2008) Image super-resolution by TV-regularization and Bregman iteration. *J Sci Comput* 37(3):367–382. <https://doi.org/10.1007/s10915-008-9214-8>
28. Martin DR, Fowlkes CC, Tal D, Malik J (2001) A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: *Proceedings of the Eighth international conference on computer vision (ICCV-01), Vancouver, British Columbia, Canada, July 7-14, 2001, vol 2*, pp 416–425. <https://doi.org/10.1109/ICCV.2001.937655>
29. McLeod M, Osborne MA, Roberts S (2018) Optimization, fast and slow: optimally switching between local and Bayesian optimization Dy J, Andreas K (eds), PMLR, Stockholm, Sweden. <http://proceedings.mlr.press/v80/mcleod18a.html>, <http://proceedings.mlr.press/v80/mcleod18a.html>
30. Pham H, Guan MY, Zoph B, Le QV, Dean J (2018) Efficient neural architecture search via parameter sharing. In: *Proceedings of the 35th international conference on machine learning, ICML 2018, Stockholm, Sweden, July 10-15, 2018*, pp 4092–4101. <http://proceedings.mlr.press/v80/pham18a.html>
31. Real E, Aggarwal A, Huang Y, Le QV (2019) Regularized evolution for image classifier architecture search. In: *Proceedings of the AAAI conference on artificial intelligence, vol 33*, pp 4780–4789
32. Schuler S, Leistner C, Bischof H (2015) Fast and accurate image upscaling with super-resolution forests. In: *IEEE conference on computer vision and pattern recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pp 3791–3799. <https://doi.org/10.1109/CVPR.2015.7299003>
33. Shi W, Caballero J, Huszár F, Totz J, Aitken AP, Bishop R, Rueckert D, Wang Z (2016) 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*, pp 1874–1883

34. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: 3rd international conference on learning representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015. Conference Track Proceedings. arXiv:1409.1556
35. Sun J, Xu Z, Shum H (2008) Image super-resolution using gradient profile prior. In: 2008 IEEE computer society conference on computer vision and pattern recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA. <https://doi.org/10.1109/CVPR.2008.4587659>
36. Timofte R, Smet VD, Gool LV (2014) A+: Adjusted anchored neighborhood regression for fast super-resolution. In: Computer vision - ACCV 2014 - 12th Asian conference on computer vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part IV, pp 111–126. https://doi.org/10.1007/978-3-319-16817-3_8
37. Timofte R, Agustsson E, Gool LV, Yang M, Zhang L, Lim B, Son S, Kim H, Nah S, Lee KM, Wang X, Tian Y, Yu K, Zhang Y, Wu Z, Dong C, Lin L, Qiao Y, Loy CC, Bae W, Yoo JJ, Han Y, Ye JC, Choi J, Kim M, Fan Y, Yu J, Han W, Liu D, Yu H, Wang Z, Shi H, Wang X, Huang TS, Chen Y, Zhang K, Zuo W, Tang Z, Luo L, Li S, Fu M, Cao L, Heng W, Bui G, Le T, Duan Y, Tao D, Wang R, Lin X, Pang J, Xu J, Zhao Y, Xu X, Pan J, Sun D, Zhang Y, Song X, Dai Y, Qin X, Huynh X, Guo T, Mousavi HS, Vu TH, Monga V, Cruz C, Egiazarian KO, Katkovnik V, Mehta R, Jain AK, Agarwalla A, Praveen CVS, Zhou R, Wen H, Zhu C, Xia Z, Wang Z, Guo Q (2017) NTIRE 2017 challenge on single image super-resolution: methods and results. In: 2017 IEEE conference on computer vision and pattern recognition workshops, CVPR Workshops 2017, Honolulu, HI, USA, July 21-26, 2017, pp 1110–1121. <https://doi.org/10.1109/CVPRW.2017.149>
38. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. *IEEE Trans Image Processing* 13(4):600–612. <https://doi.org/10.1109/TIP.2003.819861>
39. Weng Y, Zhou T, Li Y, Qiu X (2019) NAS-Unet: Neural architecture search for medical image segmentation. *IEEE Access* 7:44247–44257
40. Yan Q, Xu Y, Yang X, Nguyen TQ (2015) Single image superresolution based on gradient profile sharpness. *IEEE Trans Image Process* 24(10):3187–3202
41. Yang X, Zhou S, Cao M (2020) An approach to alleviate the sparsity problem of hybrid collaborative filtering based recommendations: the product-attribute perspective from user reviews. *MONET* 25(2):376–390. <https://doi.org/10.1007/s11036-019-01246-2>
42. Zeyde R, Elad M, Protter M (2010) On single image scale-up using sparse-representations. In: Curves and surfaces - 7th international conference, Avignon, France, June 24-30, 2010, Revised Selected Papers, pp 711–730. https://doi.org/10.1007/978-3-642-27413-8_47
43. Zhong Z, Yan J, Wu W, Shao J, Liu C (2018) Practical block-wise neural network architecture generation. In: 2018 IEEE conference on computer vision and pattern recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, pp 2423–2432. http://openaccess.thecvf.com/content_cvpr_2018/html/Zhong_Practical_Block-Wise_Neural_CVPR_2018_paper.html
44. Zoph B, Le QV (2016) Neural architecture search with reinforcement learning. arXiv:1611.01578
45. Zoph B, Vasudevan V, Shlens J, Le QV (2018) Learning transferable architectures for scalable image recognition. In:

Proceedings of the IEEE conference on computer vision and pattern recognition, pp 8697–8710

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Yu Weng received his Ph.D degree in computer science from University of Science and Technology Beijing, China, in 2010. He is currently an Associate Professor of computer science at Information Engineering Department, Minzu University of China. He has published more than 20 conference and journal papers. His current research interests include machine learning, cloud computing and distributed computing.



Zehua Chen is currently pursuing the master's degree with the Information Engineering Department, Minzu University of China. His current research interests include machine learning and computer vision.



Tianbao Zhou received the master's degree from the Information Engineering Department, Minzu University of China. He currently works for Xiaomi. His current research interests include machine learning and computer vision.