BiX-NAS: Searching Efficient Bi-directional Architecture for Medical Image Segmentation

Xinyi Wang[1,*], Tiange Xiang[1,*], Chaoyi Zhang[1], Yang Song[2], Dongnan Liu[1], Heng Huang[3,4], and Weidong Cai[1]

[1] School of Computer Science, University of Sydney, Australia
[2] School of Computer Science and Engineering, University of New South Wales, Australia
[3] Electrical and Computer Engineering, University of Pittsburgh, USA
[4] JD Finance America Corporation, Mountain View, CA, USA
{xwan2191, txia7609, dliu5812}@uni.sydney.edu.au
{chaoyi.zhang, tom.cai}@sydney.edu.au
yang.song1@unsw.edu.au
henghuanghh@gmail.com

Abstract. The recurrent mechanism has recently been introduced into U-Net in various medical image segmentation tasks. Existing studies have focused on promoting network recursion via reusing building blocks. Although network parameters could be greatly saved, computational costs still increase inevitably in accordance with the pre-set iteration time. In this work, we study a multi-scale upgrade of a bi-directional skip connected network and then automatically discover an efficient architecture by a novel two-phase Neural Architecture Search (NAS) algorithm, namely BiX-NAS. Our proposed method reduces the network computational cost by sifting out ineffective multi-scale features at different levels and iterations. We evaluate BiX-NAS on two segmentation tasks using three different medical image datasets, and the experimental results show that our BiX-NAS searched architecture achieves the state-of-the-art performance with significantly lower computational cost. Our project page is available at: https://bionets.github.io.

Keywords: Semantic Segmentation · Recursive Neural Networks · Neural Architecture Search.

1 Introduction

Deep learning based methods have prevailed in medical image analysis. U-Net [14], a widely used segmentation network, constructs forward skip connections (skips) to aggregate encoded features in encoders with the decoded ones. Recent progress has been made on the iterative inference of such architecture by exploiting the reusable building blocks. [15] proposed a recurrent U-Net that recurses

---

* Equal contributions.

a subset of paired encoding and decoding blocks at each iteration. BiO-Net [17] introduces backward skips passing semantics in decoder to the encoder at the same level. Although this recurrent design could greatly slim the network size, the computational costs still increase inevitably in accordance with its pre-set iteration time. Meanwhile, the success of multi-scale approaches [20,4] suggests the usage of multi-scale skips that fuse both fine-grained traits and coarse-grained semantics. To this end, introduction of forward/backward skips from multiple semantic scales and searching efficient aggregation of multi-scale features with low computational cost is of high interest.

Neural Architecture Search (NAS) methods automatically perceive the optimal architecture towards effective and economic performance gain. Classic evolutionary NAS methods [13,12] evolve by randomly drawing samples during searching and validating each sampled model individually. Differentiable NAS algorithms [10,3] relax the discrete search space to be continuous and delegate backpropagation to search for the best candidate. Auto-DeepLab [9] applied differentiable NAS into image segmentation tasks to determine the best operators and topologies for each building block. Concurrently, NAS-Unet [16] applied an automatic gradient-based search of cell structures to construct an U-Net like architecture. Despite the success on feature fusions at same levels, [18] introduced a multi-scale search space to endow their proposed MS-NAS with the capability of arranging multi-level feature aggregations. However, architectures searched by the above NAS algorithms only bring marginal improvements over hand-designed ones and their searching process is empirically inefficient.

In this paper, we present an efficient **multi-scale** (abstracted as 'X') NAS method, namely BiX-NAS, which searches for the optimal **bi-directional** architecture (BiX-Net) by recurrently skipping multi-scale features while discarding insignificant ones at the same time.

Our contributions are three-fold: **(1)** We study the multi-scale fusion scheme in a recurrent bi-directional manner, and present an effective two-phase Neural Architecture Search (NAS) strategy, namely BiX-NAS, that automatically searches for the optimal bi-directional architecture. **(2)** We analyze the bottleneck of the searching deficiency in classic evolution-based search algorithms, and propose a novel progressive evolution algorithm to further discover a subset of searched candidate skips and accelerate the searching process. **(3)** Our method is benchmarked on three medical image segmentation datasets, and surpasses other state-of-the-art counterparts by a considerable margin.

2   Methods

We first discuss the effectiveness of introducing multi-scale skip connections to BiO-Net as an intuitive upgrade (Sec 2.1), then we demonstrate the details of each phase of BiX-NAS (Sec 2.2), and eventually, we present the *skip fairness* principle which ensures the search fairness and efficiency (Sec 2.3).
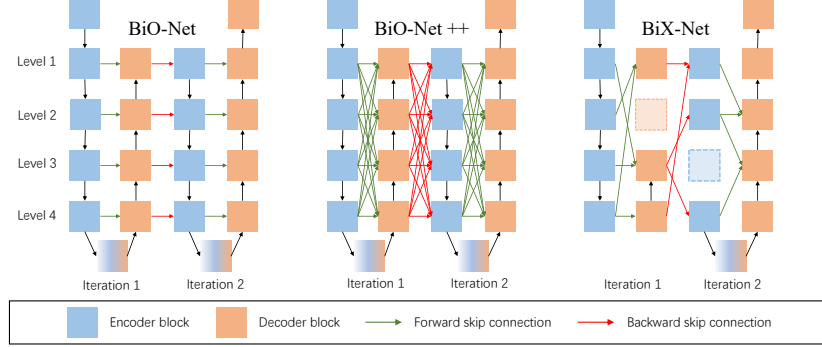
Fig. 1. Overview of BiO-Net, BiO-Net++, and BiX-Net with 4 levels and 2 iterations. Encoder and decoder blocks at the same level are reused [17].

2.1 BiO-Net++: A Multi-scale Upgrade of BiO-Net

BiO-Net [17] triggers multiple encoding and decoding phases that concatenate skipped features at the same level (Fig. 1 left). ~~We optimize the feature fusion scheme by incorporating more informative inherited priors that enhance multi~~ For better clarity, we denote (1) A sequential encoding or decoding process as an *extraction stage*. There are four extraction stages in a BiO-Net like network with two iterations. (2) Any blocks in a non-first extraction stage as *searching blocks*. To fuse multi-scale features, precedent encoded/decoded features at all levels are densely connected to every decedent decoding/encoding level through bi-directional skips. We align inconsistent spatial dimensions across different levels via bilinear resizing. The suggested BiO-Net++ is outlined in Fig. 1 middle.

Although the above design promotes multi-scale feature fusion and shrinks network size, empirically we found that such *dense* connections bring a mere marginal improvement in terms of the overall performance but with an increase of computational costs (Table 1). We are interested in seeking a *sparser* connected sub-architectures of BiO-Net++ which could not only benefit from multi-scale fusions but also ease computation burdens to the greatest extent.

2.2 BiX-NAS: Hierarchical Search for Efficient BiO-Net++

To this end, we present BiX-NAS, a two-phase search algorithm, to find a sparsely connected sub-architecture of BiO-Net++, where a trainable selection matrix is adopted to narrow down the search space for differentiable NAS in Phase1, and evolutionary NAS is introduced to progressively discover the optimal sub-architecture in Phase2. To spot an adequate candidate, dense skip connections between every pair of extraction stages are further sifted for better efficiency. Suppose there are N incoming feature streams in a desired searching block ($N = 5$ for BiO-Net++), we anticipate that in a sparser connected architecture, only $k \in [1, N - 2]$ candidate(s) of them could be accepted to such
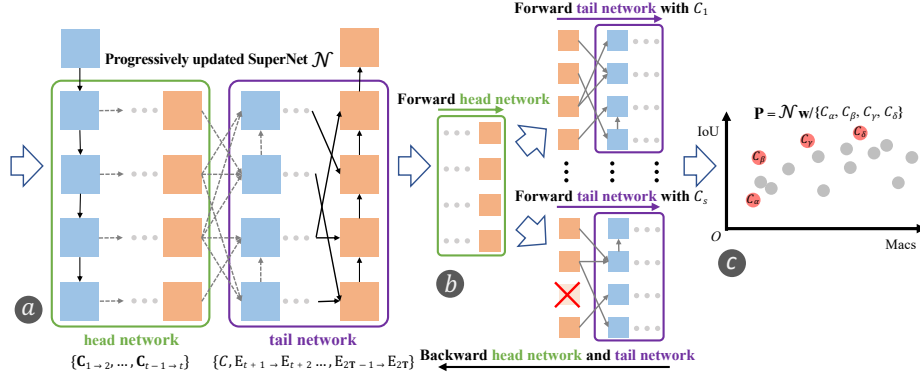
Fig. 2. Overview of progressive evolutionary search. (a) Phase1 searched Super-Net $\mathcal{N}$ can be divided into *head network* and *tail network*. **(b)** Proposed forward and backward schemes. **(c)** Only the searched skips at the Pareto front of **P** are retained.

blocks, which results in a search space of $\approx \sum_{k=1}^{(N-2)} \binom{N}{k}^{\mathbf{L}(2\mathbf{T}-1)}$ in the Super-Net BiO-Net++ with **L** levels and **T** iterations. When $\mathbf{L}=4$ and $\mathbf{T}=3$ [17], the search space expands to 5^{40}, escalating the difficulty to find the optimal sub-architectures.

Phase1: Narrowing down search space via selection matrix. To alleviate such difficulty, we determine k *candidate skips* from N incoming skips in each searching block by reducing the easy-to-spot ineffective ones. Intuitively, one-to-one relaxation parameters α [10] for each skip connection x could be registered and optimized along with the SuperNet. The skip with the highest relaxation score α is then picked as the output of $\mathbf{\Phi}(\cdot)$, such that $\mathbf{\Phi}(\cdot) = x_{\arg\max \boldsymbol{\alpha}}$, where $\mathbf{x} = \{x_1, \cdots, x_N\}$ and $\boldsymbol{\alpha} = \{\alpha_1, \cdots, \alpha_N\}$ denote the full set of incoming skips, and their corresponding relaxation scores, respectively.

However, the above formulation outputs a fixed number of skips rather than flexible ones. We continuously relax skips by constructing a learnable *selection matrix* $\mathbf{M} \in \mathbb{R}^{N \times (N-2)}$ that models the mappings between the N incoming skips and k candidates, and formulate $\mathbf{\Phi}(\cdot)$ as a fully differentiable equation below:

$$\mathbf{\Phi}(\mathbf{x}, \mathbf{M}) = Matmul(\mathbf{x}, Gumbel\_Softmax(\mathbf{M})), \tag{1}$$

where the gumbel-softmax [5] forces each of the $(N-2)$ columns of $\mathbf{M}$ to be an one-hot vector that votes for one of the N incoming skips. Our formulation generates $(N-2)$ selected skips with repetition allowed, achieving a dynamic selection of candidate skips, where the *unique ones* are further averaged out and fed into subsequent blocks. Differing from [10,9], we design $\mathbf{\Phi}(\cdot)$ to unify the forward behaviour during both network training and inference stages.

Phase2: Progressive evolutionary search. To further reduce potential redundancies among the candidate skips obtained in Phase1, we perform an additional evolutionary search to find an evolved subset of skips for better network

---

Algorithm 1 Progressive evolutionary search

Input: Iteration $\mathbf{T}$, sampling number $\mathbf{s}$, randomly initialized SuperNet weights $\mathcal{W}$,
   Phase1 searched candidate skips $\{\mathbf{C}_{1\rightarrow2}, \cdots, \mathbf{C}_{2\mathbf{T}-1\rightarrow2\mathbf{T}}\}$, criterion $\mathcal{L}$.
Output: BiX-Net with evolved skips $\{\mathbf{E}_{1\rightarrow2}, \cdots, \mathbf{E}_{2\mathbf{T}-1\rightarrow2\mathbf{T}}\}$

1: **for** $t = 2\mathbf{T} - 1, \cdots, 1$ **do**
2:    **for** $i = 1, \cdots, \mathbf{s}$ **do**
3:       **for** each searching block b **do**
4:          Randomly sample n skips from $\mathbf{C}^b_{t\rightarrow t+1}$: $C^b_i, 1 \le n \le |\mathbf{C}^b_{t\rightarrow t+1}|$.
5:       **end for**
6:    **end for**
7:    **for** data batch X, target Y **do**
8:       Forward head network with candidate skips $\mathbf{C}_{1\rightarrow2}, \cdots, \mathbf{C}_{t-2\rightarrow t-1}$.
9:       **for** $i = 1, \cdots, \mathbf{s}$ **do**
10:          Forward tail network with sampled skips $C_i, \mathbf{E}_{t+1\rightarrow t+2}, \cdots, \mathbf{E}_{2\mathbf{T}-1\rightarrow2\mathbf{T}}$.
11:          Calculate loss $l_i = \mathcal{L}(\mathrm{X}, \mathrm{Y})$
12:       **end for**
13:       Optimize $\mathcal{W}$ with the average loss $\frac{1}{\mathbf{s}}\sum_{i=1}^{\mathbf{s}} l_i$.
14:    **end for**
15:    Get Pareto front from $\{C_1, \cdots, C_\mathbf{s}\}$ and determine $\mathbf{E}_{t\rightarrow t+1}$.
16: **end for**

---

compactness and performance. Specifically, we search the candidate skips for all levels between a certain pair of adjacent extraction stages at the same time, and then progressively move to the next pair once the current search is concluded. As the connectivity of adjacent extraction stages depends on the connectivity of precedent ones, we initiate the search from the last extraction stage pair and progressively move to the first one.

The most straightforward strategies [12,13] optimize the SuperNet with each sampling skip set in a population $\mathbf{P}$ individually, and then update $\mathbf{P}$ when all training finish. However, there are two major flaws of such strategies: first, optimizing SuperNet with sampling skips individually may result in unfair outcomes; second, the searching process is empirically slow. Assuming the forward and backward of each extraction stage takes $\mathbf{I_F}$ and $\mathbf{I_B}$ time, training all $|\mathbf{P}|$ sampling architectures individually for each step takes $2\mathbf{T}|\mathbf{P}|(\mathbf{I_F} + \mathbf{I_B})$ in total.

2.3   Analysis of Searching Fairness and Deficiency

To overcome the first flaw above, we define the concept of *skip fairness* and claim that all skip search algorithms need to meet such principle.

Definition 1. *Skip Fairness.* *Let* $\mathbf{F} = \{\mathbf{f}^1, \cdots, \mathbf{f}^m\}$ *be the skipped features to any searching blocks in each sampled architecture* $\mathcal{A}^i$ *within a population* $\mathbf{P}$*. The skip fairness requires* $\mathbf{f}^1_{\mathcal{A}^1} \equiv \cdots \equiv \mathbf{f}^1_{\mathcal{A}^{|\mathbf{P}|}}, \cdots, \mathbf{f}^m_{\mathcal{A}^1} \equiv \cdots \equiv \mathbf{f}^m_{\mathcal{A}^{|\mathbf{P}|}} \ \forall \mathbf{f}^i \in \mathbf{F}, \forall \mathcal{A}^i \in \mathbf{P}$.

The above principle implies that no sharing between the [highlighted/obscured text] ... (e.g., [highlighted/obscured text]) ... is allowed. [highlighted/obscured text] Otherwise, the inconsistent incoming features would impact the search decision on the skipping topology, hence causing unexpected search unfairness. Gradient-based search algorithms (e.g. Phase1 search algorithm) meet this principle by its definition, as the same forwarded features are distributed to all candidate skips equally. However, the aforementioned straightforward strategy violates such principle due to the inconsistent incoming features produced by the individually trained architectures.

Our proposed Phase2 search algorithm meets the skip fairness by synchronizing partial forwarded features in all sampling networks. Specifically, suppose we are searching skips between the tth and $t+1^{th}$ extraction stages ($t \in [1, 2\mathbf{T}-1]$): network topology from the 1st to $t-1^{th}$ stages is fixed and the forward process between such stages can be shared. We denote such stages as *head network*. On the contrary, network topology from the tth to $2\mathbf{T}^{th}$ stages varies as the changes of different sampled skips. We then denote such unfixed stages as *tail networks*, which share the same SuperNet weights but with distinct topologies. The forwarded features of head network are fed to all sampling tail networks individually, as shown in Fig. 2. We average the losses of all tail networks, and backward the gradients through the SuperNet weights only once. Besides, our Phase2 searching process is empirically efficient and overcomes the second flaw above, as one-step training only requires $\mathbf{I_B} + \sum_{t=1}^{2\mathbf{T}-1}(t\mathbf{I_F} + (2\mathbf{T} - t)\mathbf{I_F} \cdot |\mathbf{P}|)$.

After search between each extraction stage pair completes, we follow a multi-objective selection criterion that retains the architectures at the Pareto front [19] based on both validation accuracy (IoU) and computational complexity (MACs). The proposed progressive evolutionary search details are presented in Algorithm 1 and the searched BiX-Net is shown in supplementary material Fig. 1.

3   Experiments

3.1   Datasets and Implementation Details

Two segmentation tasks across three different medical image datasets were adopted for validating our proposed method including MoNuSeg [8], TNBC [11], and CHAOS [6]. **(1)** The MoNuSeg dataset contains 30 training images and 14 testing images of size 1000×1000 cropped from whole slide images of different organs. Following [17], we extracted 512×512 patches from the corners of each image. **(2)** The TNBC dataset was used as an *extra validation dataset* [17,2], consisting of 512×512 sub-images obtained from 50 histopathology images of different tissue regions. **(3)** We also conducted 5-fold cross-validation on MRI scans from the CHAOS dataset to evaluate the generalization ability of BiX-Net on the abdominal organ segmentation task, which contains 120 DICOM image sequences from T1-DUAL (both in-phase and out-phase) and T2-SPIR with a spatial resolution of 256×256. We pre-processed the raw sequences by min-max normalization and auto-contrast enhancement.

Table 1. Comparison on MoNuSeg and TNBC with three independent runs.

Methods	MoNuSeg		TNBC		Params(M)	MACs(G)
	mIoU(%)	DICE(%)	mIoU(%)	DICE(%)		
U-Net [14]	68.2±0.3	80.7±0.3	46.7±0.6	62.3±0.6	8.64	65.83
R2U-Net [1]	69.1±0.3	81.2±0.3	60.1±0.5	71.3±0.6	9.78	197.16
BiO-Net [17]	69.9±0.2	82.0±0.2	62.2±0.4	75.8±0.5	14.99	115.67
NAS-UNet [16]	68.4±0.3	80.7±0.3	54.5±0.6	69.6±0.5	2.42	67.31
AutoDeepLab [9]	68.5±0.2	81.0±0.3	57.2±0.5	70.8±0.5	27.13	60.33
MS-NAS [18]	68.8±0.4	80.9±0.3	58.8±0.6	71.1±0.5	14.08	72.71
BiO-Net++	70.0±0.3	82.2±0.3	67.5±0.4	80.4±0.5	0.43	34.36
Phase1 searched	69.8±0.2	82.1±0.2	66.8±0.6	80.1±0.4	0.43	31.41
BiX-Net	**69.8±0.1**	**82.2±0.1**	**68.2±0.2**	**81.0±0.1**	**0.43**	**28.60**

Searching Implementation. We utilized MoNuSeg only for searching the optimal BiX-Net, and the same architecture is then transferred to all other tasks. We followed the same data augmentation strategies as in [17]. In Phase1, the BiO-Net++ SuperNet was trained with a total of 300 epochs with a base learning rate of 0.001 and a decay rate of $3e^{-3}$. The network weights and the selection matrices were optimized altogether with the same optimizer, rather than optimized separately [10]. Our Phase1 searching procedure took only 0.09 GPU-Day. In Phase2, there were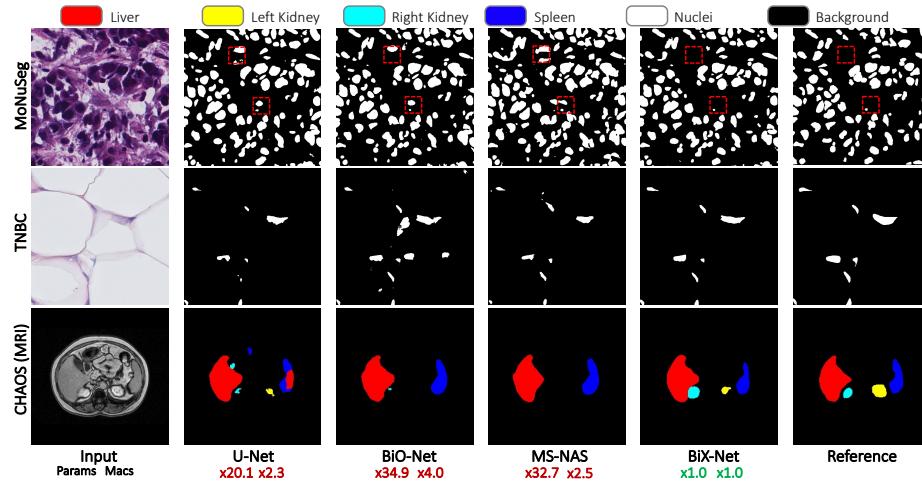 total 5 searching iterations when $\mathbf{T} = 3$. ~~In each iteration,~~ ~~fixed architectures are fine-tuned for only a total of 40 epochs. We retrained 15~~ ~~different architectures for only 40 epochs during Phase2, each required 40 epochs~~ ~~training from a base learning rate of 0.001 and decreased by 10 times every 10~~ ~~epochs. Every GPU ~~only~~ handles 5 candidate architectures. The candidate~~ ~~with highest mIoU on the Best set after Our Phase2 searching procedure took~~ ~~0.07 GPU-Day.~~

Retraining Implementation. Adam optimizer [7] was used across all experiments to minimize cross entropy loss in both network searching and retraining. Batch size was set to be 2 for MoNuSeg and TNBC, and 16 for CHAOS. We retrained the constructed BiX-Net and all competing models with the same implementation as Phase1, which was identical across all experiments for fair comparisons. Mean intersection of Union (mIoU) and Dice Coefficient (DICE) were reported to evaluate accuracy while Multiplier Accumulator (MACs) was reported to measure computational complexity.

3.2 Experimental Results

Nuclei Segmentation. Our method was compared to the vanilla U-Net [14], state-of-the-art recurrent U-Net variants [1,17], and homogeneous state-of-the-art NAS searched networks [16,9,18]. All models were trained from scratch with final results reported as the average of three independent runs. Table 1 shows that our plain BiX-Net outperforms the state-of-the-art NAS counterparts considerably, and achieves on par results to BiO-Net [17] with significantly lower

Table 2. Comparison on CHAOS (MRI) with 5-fold cross validation.

Methods	Liver		Left Kidney		Right Kidney		Spleen	
	mIoU(%)	DICE(%)	mIoU(%)	DICE(%)	mIoU(%)	DICE(%)	mIoU(%)	DICE(%)
U-Net	78.1±2.0	86.8±1.8	61.3±1.1	73.8±1.2	63.5±1.1	76.2±1.1	62.2±2.1	74.4±2.3
BiO-Net	85.8±2.0	91.7±1.8	75.7±1.1	85.1±1.2	78.2±1.0	87.2±1.1	73.2±2.3	82.8±2.3
NAS-UNet	79.1±1.8	87.2±1.8	65.5±1.5	75.0±1.3	66.2±1.2	77.7±1.0	64.1±1.3	75.8±1.6
AutoDeepLab	79.8±1.9	88.1±1.8	66.7±1.6	75.0±1.7	61.9±0.9	75.7±1.1	63.9±1.2	75.5±1.4
MS-NAS	72.6±2.3	82.6±2.1	71.0±1.3	81.9±1.3	70.1±1.9	81.1±1.8	62.5±2.1	74.0±2.3
BiX-Net	**82.6±1.5**	**89.8±1.5**	**71.0±1.0**	**82.1±1.1**	**71.9±0.8**	**82.7±1.0**	**66.0±1.7**	**76.5±2.0**

Fig. 3. Qualitative comparison of different segmentation results. Relative parameter (left) and computation overheads (right) compared to BiX-Net are also shown.

network complexity. Noteworthy, our results are much higher than all others on TNBC, which indicates superior generalization ability. Qualitative comparisons on all datasets are shown in Fig. 3.

Multi-class Organ Segmentation. CHAOS challenge aims at the precise segmentation of four abdominal organs separately: liver, left kidney, right kidney, and spleen in a CT or MRI slice. Instead of training networks on each class as several independent binary segmentation tasks [18], we reproduced all models to output the logits for all classes directly. Similar to the nuclei segmentation, Table 2 indicates that BiX-Net achieves the best performance of all classes among state-of-the-art NAS searched networks [16,9,18] with much lower computational complexity. Although the hand-crafted BiO-Net outperforms all comparison methods, it suffers from the computation burdens, which are a 4.0 times of computational complexity, and a 34.9 times of trainable parameter number. Additionally, BiX-Net produces much better segmentation mask when all organs are presented in a single slice (Fig. 3).

4 Conclusion

In this work, we proposed an efficient two-phase NAS algorithm that searches for bi-directional multi-scale skip connections between encoder and decoder, namely BiX-NAS. We first follow differentiable NAS with a novel selection matrix to narrow down the search space. An efficient progressive evolutionary search is then proposed to further reduce skip redundancies. Experimental results on various segmentation tasks show that the searched BiX-Net surpasses state-of-the-art NAS counterparts with considerably fewer parameters and computational costs.

References

1. Alom, M.Z., Yakopcic, C., Taha, T.M., Asari, V.K.: Nuclei segmentation with recurrent residual convolutional neural networks based u-net (r2u-net). In: IEEE National Aerospace and Electronics Conference. pp. 228–233. IEEE (2018)
2. Graham, S., Vu, Q.D., Raza, S.E.A., Azam, A., Tsang, Y.W., Kwak, J.T., Rajpoot, N.: Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images. Medical Image Analysis **58**, 101563 (2019)
3. Guo, Z., Zhang, X., Mu, H., Heng, W., Liu, Z., Wei, Y., Sun, J.: Single path one-shot neural architecture search with uniform sampling. In: European Conference on Computer Vision (ECCV). pp. 544–560. Springer (2020)
4. Huang, H., Lin, L., Tong, R., Hu, H., Zhang, Q., Iwamoto, Y., Han, X., Chen, Y.W., Wu, J.: UNet 3+: A full-scale connected unet for medical image segmentation. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1055–1059. IEEE (2020)
5. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. In: International Conference on Learning Representations (ICLR) (2017)

6. Kavur, A.E., Gezer, N.S., Barış, M., Aslan, S., Conze, P.H., Groza, V., Pham, D.D., Chatterjee, S., Ernst, P., Özkan, S., et al.: CHAOS challenge-combined (CT-MR) healthy abdominal organ segmentation. Medical Image Analysis **69**, 101950 (2021)

7. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR) (2015)

8. Kumar, N., Verma, R., Sharma, S., Bhargava, S., Vahadane, A., Sethi, A.: A dataset and a technique for generalized nuclear segmentation for computational pathology. IEEE Transactions on Medical Imaging **36**(7), 1550–1560 (2017)

9. Liu, C., Chen, L.C., Schroff, F., Adam, H., Hua, W., Yuille, A.L., Fei-Fei, L.: Auto-DeepLab: Hierarchical neural architecture search for semantic image segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 82–92 (2019)

10. Liu, H., Simonyan, K., Yang, Y.: DARTS: Differentiable architecture search. In: International Conference on Learning Representations (ICLR) (2019)

11. Naylor, P., Laé, M., Reyal, F., Walter, T.: Segmentation of nuclei in histopathology images by deep regression of the distance map. IEEE Transactions on Medical Imaging **38**(2), 448–459 (2018)

12. Real, E., Aggarwal, A., Huang, Y., Le, Q.V.: Regularized evolution for image classifier architecture search. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 4780–4789 (2019)

13. Real, E., Moore, S., Selle, A., Saxena, S., Suematsu, Y.L., Tan, J., Le, Q.V., Kurakin, A.: Large-scale evolution of image classifiers. In: International Conference on Machine Learning. pp. 2902–2911. PMLR (2017)

14. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI). pp. 234–241. Springer (2015)

15. Wang, W., Yu, K., Hugonot, J., Fua, P., Salzmann, M.: Recurrent U-Net for resource-constrained segmentation. In: The IEEE International Conference on Computer Vision (ICCV) (2019)

16. Weng, Y., Zhou, T., Li, Y., Qiu, X.: NAS-Unet: Neural architecture search for medical image segmentation. IEEE Access **7**, 44247–44257 (2019)

17. Xiang, T., Zhang, C., Liu, D., Song, Y., Huang, H., Cai, W.: BiO-Net: Learning recurrent bi-directional connections for encoder-decoder architecture. In: International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI). pp. 74–84. Springer (2020)

18. Yan, X., Jiang, W., Shi, Y., Zhuo, C.: MS-NAS: Multi-scale neural architecture search for medical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI). pp. 388–397. Springer (2020)

19. Yang, Z., Wang, Y., Chen, X., Shi, B., Xu, C., Xu, C., Tian, Q., Xu, C.: CARS: Continuous evolution for efficient neural architecture search. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1829–1838 (2020)

20. Zhou, Z., Siddiquee, M.M.R., Tajbakhsh, N., Liang, J.: UNet++: A nested u-net architecture for medical image segmentation. In: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support (DLMIA), pp. 3–11. Springer (2018)